AutoMap User's Guide 2012

Kathleen M. Carley, Dave Columbus, and Ariel Azoulay

June 11, 2012
CMU-ISR-12-106

Institute for Software Research
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Center for the Computational Analysis of Social and Organization Systems
CASOS technical report

This report/document supersedes CMU-ISR-11-108R
"AutoMap User's Guide 2011", June 2011

This work was supported, in part, by the Office of Naval Research - MURI - A Structural Approach to the Incorporation of Cultural Knowledge in Adaptive Adversary Models (N000140811186), Office of Naval Research - Rules of Engagement (N00014-06-1-0104), Office of Naval Research - Expansion to DNA Merchant Marine Traffic (N00014-06-1-0104), SORASCS - Architecture to Support Socio-Cultural Modeling (N000140811223); Office of Naval Research - CATNET: Competitive Adaptation in Terrorist Networks (N00014-09-1-0667); the AirForce Office of Sponsored Research - MURI with GMU - Cultural Modeling of the Adversary (FA9550-05-1-0388); the Defense Threat Reduction Agency - Remote Capabilities Assessment (HDTRA11010102); the Army Research Office - Learned Resiliency: Secure Multi-Level Systems (W911NF-09-1-0273); the Army Research Institute - Improved Data Extraction and Assessment for Dynamic Network Analysis (W91WAW-07-C-0063); and Netanomics. Additional support was provided by the center for Computational Analysis of Social and Organizational Systems (CASOS) at Carnegie Mellon University. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Department of Defense (DoD), the Office of Naval Research (ONR), the Air Force Office of Sponsored Research (AFOSR), the Defense Threat Reduction Agency (DTRA), the Army Research Office (ARO), the Army Research Institute (ARI) or the U.S. government.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
11 JUN 2012		**00-00-2012 to 00-00-2012**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
AutoMap User's Guide 2012	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Carnegie Mellon University,Institute for Software Research,School of Computer Science,Pittsburgh,PA,15213**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT
AutoMap is an advanced text mining system. It operates in 4 modes. First, it can do classical content analysis; i.e. concepts and their frequency. Second, it extracts the semantic network; i.e. concepts and their relation to each other. Third, it cross classifies the concepts into their ontological categories such as agents and locations which results in meta-network. This includes, e.g. the social network. Fourth, it utilizes post processing to infer various aspects of sentiment.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**255**	

Keywords: Semantic Network Analysis, Dynamic Network Analysis, Mental Modes, Social Networks, AutoMap

Abstract

AutoMap is an advanced text mining system. It operates in 4 modes. First, it can do classical content analysis; i.e. concepts and their frequency. Second, it extracts the semantic network; i.e. concepts and their relation to each other. Third, it cross classifies the concepts into their ontological categories such as agents and locations which results in meta-network. This includes, e.g. the social network. Fourth, it utilizes post processing to infer various aspects of sentiment.

Table of Contents

AutoMap 3 Overview **1**
Resources **3**
Glossary 3
GUI Quickstart 13
Simple Tutorials **16**
Content Analysis to Semantic Network **20**
Interface Details **23**
Script Quickstart 25
AM3Script Tags Details **29**
Simple Tutorials **42**
Non-English Fonts 47
Java Licenses 48
Content Section **58**
Anaphora 59
Bi-Grams 60
Concept Lists 64
Data Selection 65
Delete Lists 67
Delete Lists 69
Text Encoding 75
Text Encoding Table 77
File Formats 79
Format Case 80
Master Format 81
Meta-Network Thesaurus 81
Named Entities 82
Networks 82
Ontology 85
Parts of Speech 87
Relation Types 90
Semantic Lists 95
Semantic Networks 97
Process Sequencing 99

Stemming 100
Text Formats 103
Text Properties 104
Thesauri, General 105
Thesauri, MetaNetwork 108
Thesaurus Content Only 110
Threshold, Global and Local 113
Union 117
Union Concept List 118
Window Size 122
GUI Section 123
The GUI (Graphic User Interface) 124
File Menu 127
File Menu-Conversions 129
File Menu-Save 131
Edit Menu 132
Edit-Preferences 133
CEMap 136
Extractors Menu 138
Preprocessing Menu 142
Text Cleaning Menu 143
Preprocessing Menu 144
Refinement Menu 146
Generate Menu 148
Generate-Parts Of Speech 150
Generate-Concept Lists 151
Generate-Semantic Networks 153
Generate-Meta-Networks 154
Generate-Thesaurus Suggestion 156
Generate-Generalization Thesauri 158
Procedures 160
Procedures-Master Thesauri 161
Procedures-Concept List 167
Procedures-Thesauri 168

Procedures-Delete Lists 171
Procedures-DyNetML 172
Tools Menu 174
Tools 176
Delete List Editor 177
Thesauri Editor 180
Attribute Editor 185
Concept List Viewer 188
Table Viewer 192
XML Viewer 194
Tagged Text Viewer 197
Script Runner 202
Compare Color Chart 208
Text Partitioner 208
Script 210
AM3Script Notes 210
AM3Script Tags 213
AM3Script Tags-Script 214
AM3Script Tags-Extractors 214
AM3Script Tags-PreProcessing 216
AM3Script Tags-Processing 218
AM3Script Tags-Procedures 222
AM3Script Tags-Post-Processing 225
DOS Commands 227
Run Script Anywhere 231
Data-to-Model 233
Basic Model 234
Refined Model 241
Advanced Model 242
Analysis 242
References 243

AutoMap 3 Overview

An Overview

AutoMap is text analysis software that implements the method of Network Text Analysis, specifically Semantic Network Analysis. Semantic analysis extracts and analyzes links among words to model an author's **mental map** as a network of links. Automap also supports Content Analysis.

Coding in AutoMap is computer-assisted; the software applies a set of coding rules specified by the user in order to code the texts as networks of concepts. Coding texts as maps focuses the user on investigating meaning among texts by finding relationships among words and themes.

The coding rules in AutoMap involve text pre-processing and statement formation, which together form the coding scheme. Text pre-processing condenses data into concepts, which capture the features of the texts relevant to the user. Statement formation rules determine how to link concepts into statements.

Network Text Analysis (NTA)

Network Text Analysis theory is based on the assumption that language and knowledge can be modeled as networks of words and relations. NTA encodes links among words to construct a network of linkages. Specifically, this method analyzes the existence, frequencies, and covariance of terms and themes, thus subsuming classical Content Analysis.

Social Network Analysis (SNA)

Social Network Analysis (Wasserman & Faust, 1994) is a scientific area focused on the study of relations, often defined as social networks. In its basic form, a social network is a network where the nodes are people and the relations (also called links or ties) are a form of connection such as friendship. Social Network Analysis (Wasserman & Faust, 1994) takes graph theoretic ideas and applies them to the social world. The term "social network" was first coined in 1954 by J. A. Barnes (see: Class and Committees in a Norwegian Island Parish). Social network

analysis (Wasserman & Faust, 1994) is also called network analysis, structural analysis, and the study of human relations. SNA is often referred to as the science of **connecting the dots**.

Today, the term Social Network Analysis (Wasserman & Faust, 1994) is used to refer to the analysis of any network such that all the nodes are of one type (e.g., all people, or all roles, or all organizations), or at most two types (e.g., people and the groups they belong to). The metrics and tools in this area, since they are based on the mathematics of graph theory, are applicable regardless of the type of nodes in the network or the reason for the connections.

For most researchers, the nodes are actors. As such, a network can be a cell of terrorists, employees of global company or simply a group of friends. However, nodes are not limited to actors. A series of computers that interact with each other or a group of interconnected libraries can also comprise a network.

Semantic Network Analysis

In map analysis, a concept is a single idea, or ideational kernel, represented by one or more words. Concepts are equivalent to nodes in Social Network Analysis (SNA) (Wasserman & Faust, 1994). The link between two concepts is referred to as a statement, which corresponds with an edge in SNA. The relation between two concepts can differ in strength, directionality, and type. The union of all statements per texts forms a semantic map. Maps are equivalent to networks.

Dynamic Network Analysis

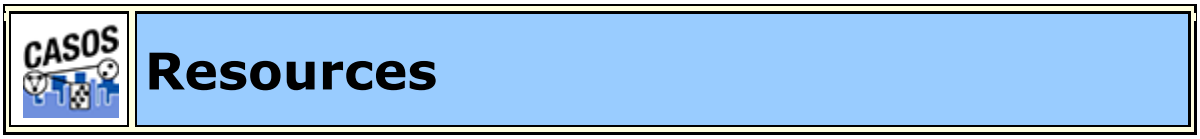
Dynamic Network Analysis (DNA) is an emergent scientific field that brings together traditional social network analysis (SNA) (Wasserman & Faust, 1994), link analysis (LA) and multi-agent systems (MAS). There are two aspects of this field. The first is the statistical analysis of DNA data. The second is the utilization of simulation to address issues of network dynamics. DNA networks vary from traditional social networks in that there are larger dynamic multi-mode, multi-plex networks, and may contain varying levels of uncertainty.

DNA statistical tools are generally optimized for large-scale networks and simultaneously admit the analysis of multiple

networks in which there are multiple types of entities (multi-entities) and multiple types of links (multi-plex). In contrast, SNA statistical tools focus on single or at most two mode data and facilitate the analysis of only one type of link at a time.

Because they have measures that use data drawn from multiple networks simultaneously, DNA statistical tools tend to provide more measures to the user. From a computer simulation perspective, entities in DNA are like atoms in quantum theory: they can be, though need not be, treated as probabilistic. Whereas entities in a traditional SNA model are static, entities in a DNA model have the ability to learn. Properties change over time; entities can adapt. For example, a company's employees can learn new skills and increase their value to the network, or one terrorist's death forces three more to improvise. Change propagates from one entity to the next and so on. DNA adds the critical element of a network's evolution to textual analysis and considers the circumstances under which change is likely to occur.

4 JAN 11



Description

Contained within these pages are resources useful in using AutoMap.

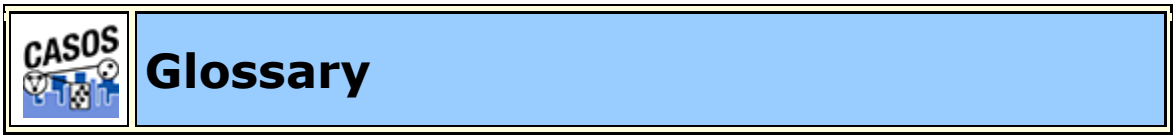
Glossary of terms used in describing AutoMap.

GUI Quickstart guide.

Script Quickstart guide.

Non-English Font web sites.

13 OCT 09



Adjacency Network : A Network that is a square actor-by-actor (i=j) network where the presence of pairwise links are recorded as elements. The main diagonal, or self-tie of an adjacency network is often ignored in network analysis.

Aggregation : Combining statistics from different nodes to higher nodes.

Algorithm : A finite list of well-defined instructions for accomplishing some task that, given an initial state, will terminate in a defined end-state.

Attribute : Indicates the presence, absence, or strength of a particular connection between nodes in a Network.

Betweenness : Degree an individual lies between other individuals in the network; the extent to which an node is directly connected only to those other nodes that are not directly connected to each other; an intermediary; liaisons; bridges. It is the number of nodes a given node is indirectly connected to via its direct links.

Betweenness Centrality : High in betweenness but not degree centrality. This node connects disconnected groups, like a Go-between.

Bigrams : Bigrams are groups of two written letters, two syllables, or two words, and are very commonly used as the basis for simple statistical analysis of text.

Bimodal Network : A network most commonly arising as a mixture of two different unimodal networks.

Binarize : Divides your data into two sets; zero or one.

Bipartite Graph : Also called a bigraph. It's a set of nodes decomposed into two disjoint sets such that no two nodes within the same set are adjacent.

BOM : A byte order mark (BOM) consists of the character code U+FEFF at the beginning of a data stream, where it can be used as a signature defining the byte order and encoding form, primarily of unmarked plaintext files. Under some higher level

protocols, use of a BOM may be mandatory (or prohibited) in the Unicode data stream defined in that protocol.

Centrality : The nearness of an node to all other nodes in a network. It displays the ability to access information through links connecting other nodes. The closeness is the inverse of the sum of the shortest distances between each node and every other node in the network.

Centralization : Indicates the distribution of connections in the employee communication network as the degree to which communication and/or information flow is centralized around a single agent or small group.

Classic SNA density : The number of links divided by the number of possible links not including self-reference. For a square network, this algorithm* first converts the diagonal to 0, thereby ignoring self-reference (a node connecting to itself) and then calculates the density. When there are N nodes, the denominator is (N*(N-1)). To consider the self-referential information, use general density.

Clique : A sub-structure that is defined as a set of nodes where every node is connected to every other node.

Clique Count : The number of distinct cliques to which each node belongs.

Closeness : Node that is closest to all other Nodes and has rapid access to all information.

Clustering coefficient : Used to determine whether or not a graph is a small-world network.

Cognitive Demand : Measures the total amount of effort expended by each agent to do its tasks.

Collocation : A sequence of words or terms which co-occur more often than would be expected by chance.

Column Degree : see Out Degree*.

Complexity : Complexity reflects cohesiveness in the organization by comparing existing links to all possible links in all four networks (employee, task, knowledge and resource).

Concor Grouping : Concor recursively splits partitions and the user selects n splits. (n splits -> 2n groups). At each split it divides the nodes based on maximum correlation in outgoing connections. Helps find groups with similar roles in networks, even if dispersed.

Congruence : The match between a particular organizational design and the organization's ability to carry out a task.

Count : The total of any part of a Meta-Network row, column, node, link, isolate, etc.

CSV : "Comma Separated Value". A common file structure used in database programs for formatting output data.

Degree : The total number of links to other nodes in the network.

Degree Centrality : Node with the most connections. (e.g. In the know). Identifying the sources for intel helps in reducing information flow.

Density :

- **Binary Network :** The proportion of all possible links actually present in the Network.
- **Value Network :** The sum of the links divided by the number of possible links. (e.g. the ratio of the total link strength that is actually present to the total number of possible links).

Dyad : Two nodes and the connection between them.

Dyadic Analysis : Statistical analysis where the data is in the form of ordered pairs or dyads. The dyads in such an analysis may or may not be for a network.

Dynamic Network Analysis : Dynamic Network Analysis (DNA) is an emergent scientific field that brings together traditional

Social Network Analysis* (SNA), Link Analysis* (LA) and multi-agent systems (MAS).

DyNetML : DynetML is an xml based interchange language for relational data including nodes, ties, and the attributes of nodes and ties. DyNetML is a universal data interchange format to enable exchange of rich social network data and improve compatibility of analysis and visualization tools.

Endain : Data types longer than a byte can be stored in computer memory with the most significant byte (MSB) first or last. The former is called big-endian, the latter little-endian. When data are exchange in the same byte order as they were in the memory of the originating system, they may appear to be in the wrong byte order on the receiving system. In that situation, a BOM would look like 0xFFFE which is a non-character, allowing the receiving system to apply byte reversal before processing the data. UTF-8 is byte oriented and therefore does not have that issue. Nevertheless, an initial BOM might be useful to identify the data stream as UTF-8.

Entropy : The formalization of redundancy and diversity. Thus we say that Information Entropy (H) of a text document (X) where probability p of a word x = ratio of total frequency of x to length (total number of words) of a text document.

General density : The number of links divided by the number of possible links including self-reference. For a square network, this algorithm* includes self-reference (an node connecting to itself) when it calculates the density. When there are N nodes, the denominator is (N*N). To ignore self-referential information use classic SNA* density.

Hidden Markov Model : A statistical model in which the system being modeled is assumed to be a Markov process with unknown parameters, and the challenge is to determine the hidden parameters from the observable parameters.

Homophily : (e.g., love of the same) is the tendency of individuals to associate and bond with similar others.

- **Status homophily** means that individuals with similar social status characteristics are more likely to associate with each other than by chance.

- **Value homophily** refers to a tendency to associate with others who think in similar ways, regardless of differences in status.

In-Degree : The sum of the connections leading to an node from other nodes. Sometimes referred to row degree.

Influence network : A network of hypotheses regarding task performance, event happening and related efforts.

Isolate : Any node which has no connections to any other node.

Link : A specific relation among two nodes. Other terms also used are tie and link.

Link Analysis : A scientific area focused on the study of patterns emerging from dyadic observations. The relationships are typically a form of co-presence between two nodes. Also multiple dyads that may or may not form a network.

Main Diagonal : in a square network this is the conjunction of the rows and cells for the same node.

Network Algebra : The part of algebra that deals with the theory of networks.

Meta-Network : A statistical graph of correlating factors of personnel, knowledge, resources and tasks. These measures are based on work in social networks, operations research, organization theory, knowledge management, and task management.

Morpheme : A morpheme is the smallest meaningful unit in the grammar of a language.

Multi-node : More than one type of node (people, events, locations, etc.).

Multi-plex : Network where the links are from two or more relation classes.

Multimode Network : Where the nodes are in two or more node classes.

Named Entity List (NEL) : A list of ngrams that are thought to refer to specific people, organizations, or locations.

Named-Node Recognition : An Automap feature that allows you to retrieve proper names (e.g. names of people, organizations, places), numerals, and abbreviations from texts.

Neighbors : Nodes that share an immediate link to the node selected.

NEL (project original) : This is the named entity list auto-generated by AutoMap with AutoMap guesses as to ontology class.

> **NOTE :** It may contain entities that are not true named entities and the classification may be wrong.
>
> **NOTE :** The size of this list is constant for a given version of automap and depends only on the tools in automap.

NEL (project unclassified) : This is what remains of the NEL (project original) after named entities from the standard thesauri are removed and after named entities classified by a human are removed.

> **NOTE :** The size of this list will shrink each time the NEL (project original) is processed with a new standard thesauri and new project specific classifications of named entities.

In general, most users will do 2 to 3 passes of cleaning the NEL resulting in "additional project thesauri." If all these additions plus the standard are applied to NEL (project original) or if just the most recent addition is applied to NEL (project unclassified), the resulting NEL (project unclassified) and NEL (project classified) should be identical.

> **NOTE :** Not all terms may end up being classified.

NEL (project classified) : This is the set of NEL drawn from the project corpus that are classified by ontological category and have been checked for accuracy.

> **NOTE :** This includes all of n-grams in the project corpus that according to the standard thesauri are NEL.

Checking for accuracy means either it was classified by the standard thesauri or a project user classified the term. Standard thesuari should be applied first.

NOTE : The size of the NEL (project classified) should increase as more terms from the NEL (project unclassified) are classified.

NOTE : After the project is done, a CASOS person should determine which if any of the project NEL should get added to the standard thesauri.

Network : Set of links among nodes. Nodes may be drawn from one or more node classes and links may be of one or more relation classes.

Newman Grouping : Finds unusually dense clusters, even in large networks.

Nodes : General things within an node class (e.g. a set of actors such as employees).

Node Class : The type of items we care about (knowledge, tasks, resources, agents).

Node Level Metric : is one that is defined for, and gives a value for, each node in a network. If there are x nodes in a network, then the metric is calculated x times, once each for each node. Examples are Degree Centrality*, Betweenness*, and Cognitive Demand*.

Node Set : A collection of nodes that group together for some reason.

ODBC : (O)pen (D)ata (B)ase (C)onnectivity is an access method developed by the SQL Access group in 1992 whose goal was to make it possible to access any data from any application, regardless of which database management system (DBMS) is handling the data.

Ontology : "The Specifics of a Concept". The group of nodes, resources, knowledge, and tasks that exist in the same domain and are connected to one another. It's a simplified way of viewing the information.

Organization : A collection of networks.

Out-Degree : The sum of the connections leading out from an node to other nodes. This is a measure of how influential the node may be. Sometimes referred to as column degree.

Pendant : Any node which is only connected by one link. They appear to dangle off the main group.

Project : The thing you are working on. This is generally associated with a research question.

Project corpus : The set of texts used in a specific project. These often exist in raw and in cleaned form. The cleaned form would be just .txt files.

Random Graph : One tries to prove the existence of graphs with certain properties by assigning random links to various nodes. The existence of a property on a random graph can be translated to the existence of the property on almost all graphs using the famous Szemerédi regularity lemma*.

Reciprocity : The percentage of nodes in a graph that are bi-directional.

Redundancy : Number of nodes that access to the same resources, are assigned the sametask, or know the same knowledge. Redundancy occurs only when more than one agent fits the condition.

Relation : The way in which nodes in one class relate to nodes in another class.

Row Degree : see In Degree*.

Semantic Network : Often used as a form of knowledge representation. It is a directed graph consisting of vertices, which represent concepts, and links, which represent semantic relations between concepts.

Social Network Analysis : The term Social Network Analysis (or SNA) is used to refer to the analysis of any network such that all the nodes are of one type (e.g., all people, or all roles, or

all organizations), or at most two types (e.g., people and the groups they belong to).

Specific Entity : The name by which the person, organization or location is commonly referred to that identifies them as distinct from a generic entity. For example, John Doe is specific man is generic.

Stemming : Stemming detects inflections and derivations of concepts in order to convert each concept into the related morpheme.

tfidf : Term Frequency/Inverse Document Frequency helps determine a word's importance in the corpus. **tf (Term Frequency)** is the importance of a term within a document. **idf (Inverse Document Frequency** is the importance of a term within the corpus.

$$tf = \frac{cumulative\ occurance\ of\ term\ x\ in\ document\ y}{total\ number\ of\ terms\ in\ documet\ y}$$

$$idf = \frac{log\,(total\ number\ of\ documents\ in\ corpus)}{total\ number\ of\ documents\ containing\ term\ x}$$

tfidf = tf * idf

Useful when creating a General Thesaurus.

Thesaurus : A list which associates multiple abstract concepts with more common concepts.

- **Generalization Thesaurus :** Typically a two-columned collection that associates text-level concepts with higher-level concepts. The text-level concepts represent the content of a data set, and the higher-level concepts represent the text-level concepts in a generalized way.
- **Meta-Network Thesaurus :** Associates text-level concepts with meta-network categories.

Sub-Matrix Selection : The Sub-Matrix Selection denotes which Meta-Network Categories should be retranslated into concepts used as input for the meta-network thesaurus.

Topology : The study of the arrangement or mapping of the elements (links, nodes, etc.) of a network, especially the physical (real) and logical (virtual) interconnections between nodes.

Unimodal networks : These are also called square networks because their adjacency network* is square; the diagonal is zero diagonal because there are no self-loops*.

Windowing : A method that codes the text as a map by placing relationships between pairs of Concepts that occur within a window. The size of the window can be set by the user.

12 JUN 09

GUI Quickstart

AutoMap is a natural language processing system. It is used as a means to understand text, or to process text to be used in conjunction with other tools such as the CASOS *ORA program. Some of the ways in which AutoMap is used:

1. To extract a metanetwork representation of a dynamic/social network as expressed in text.
2. To extract a semantic network to understand the relationships between concepts in texts.
3. To clean and process text files for example by removing symbols and numbers, deleting unnecessary words, and stemming.
4. To identify concepts and the frequency of concepts appearing in texts.

Description

The AutoMap GUI (Graphical User Interface) contains access to AutoMap's features via the menu items and shortcut buttons. The purpose of the GUI is to aid in the exploration of processing steps. Users will be able to understand the impact of processing parameters and processing order.

The processing of an extensive collection of texts is best done using the script version of AutoMap. The same processing steps available in the AutoMap GUI are available in the AutoMap Script.

Guide Roadmap

A. Interface Overview

B. Tutorial 1: Creating Concept and Union Concept List

C. Tutorial 2: Using Delete Lists

D. Tutorial 3: Content Analysis to Semantic Network

E. Interface Details

The User Interface Overview

The Pull Down Menus

The **Text Display Window** displays the text file as it appears based on the preprocessing that has been applied to it. The **File Navigation Buttons** allow you to move between individual text files. The **Filename Box** will identify the name of the currently displayed text file.

The **Message Window** will provide feedback. The **Quick Launch Buttons** are the most commonly used menu commands, placed in the main window for quick access.

The **File Menu** contains loading and saving commands, and exit, to quit the AutoMap program.

The **Edit Menu** contains configuration options.

The **Preprocess Menu** contains commands that will modify the text file. These commands may be applied in any order. The result of the preprocessing is displayed in the Text Display Window, with the name of the preprocessing step displayed in the Preprocess /Order Window.

The **Generate Menu** contains commands for generating end results. The output of these commands may be

created to be used as input to other programs. For instance, a generated MetaNetwork DyNetML file can be used as input to *ORA for analysis.

The **Tools Menu** contains launchable external tools. These tools are provided to aid in the editing of supplemental files or the viewing of end results. AutoMap uses standard file formats such as text (.txt), comma separated value (.csv) or XML (.xml) in order to provide maximum interaction with other tools.

The Help Menu contains the AutoMap help system.

Before You Begin

AutoMap is a system that starts with text files. Before being able to use the features of AutoMap, it is necessary to have text to process. This text can be obtained from email, news articles, publications, web pages, or text typed in using a text editor.

AutoMap will process all text (.txt) files in a directory. It is not necessary to combine text into a single file. Some larger text files can be split into smaller text files to do analysis of sections individually.

You will be prompted for the location of where to store the files that are the results of your processing. Many people will create a folder to keep the text files and all of the results. In this work folder, create a subfolder to store the original texts and additional subfolders to store the results you will generate.

For example, if we are interested in only creating concept lists from our texts, we can create the following file structure:

C:\Mike\working

C:\Mike\working\texts

C:\Mike\working\concepts

When generating a concept list, be sure to navigate to the appropriate folder, such as C:\Mike\working\concepts folder in our example, to store the results.

Simple Tutorials

Creating Concept & Union Concept Lists

Description

Concept Lists & Union Concept Lists compile lists based on individual and multiple files giving their frequency. A Concept List collects concepts in one file only. Union Concept Lists collect concepts from all currently loaded files.

Step 1: Load Text Files

From the Pull Down Menu select ***File => Select Input Directory***. Navigate to a directory with your text files and click Select.

Step 2: Create a Concept List

From the Pull Down Menu select ***Generate => Concept List***. Navigate to a directory to save the list and click **Select**. If you have other files in that directory, you will be alerted that some files may be overwritten. As long as you did not add or remove input files from a previous run there is no problem as the previous concept list files will be overwritten with the new concept list files. The file name will be the same as the original text file, substituting the.txt for.csv. For instance mike.txt as an input text file will create a concept list file named mike.csv.

AutoMap will ask if you want to generate a **Union Concept List**. It is a good idea to create this list. All files in the directory you select to save your concept lists in will be used to create the union concept list. If you have old concept lists in there not from the current run, they will also be used.

Viewing a Concept List

From the Pull Down Menu select ***Tools => Concept List Viewer***. From the Viewer Pull Down Menu select ***File => Open File***. Navigate to the directory where your Concept Lists are stored and select one and click **Open**. If a **Concept List** is chosen only the concepts from one file are displayed. If a **Union Concept List** is chosen it will display concepts from all files. As the concept lists are saved in a standard.csv format, you can also view them in a text editor or a spreadsheet program such as Microsoft Excel.

Creating a Delete List

From the viewer menu you can create a Delete List by placing a check mark in the **Selected** columns then from the Pull Down Menu select ***File => Save as Delete List***. Navigate to the directory, type in a new file name, and click **Open** to save your new Delete List.

Comparing Files

You can also compare the currently loaded file with another using ***File => Compare File***. Navigate to the file to compare the first file with and click **Open**.

AutoMap will color code the concepts: no color means the information is the same in both the original and compared files, **red** means the concept was in the original but not in the compared file, **green** means the concept was not in the original but is in the compared file, and **yellow** the concepts are the same but the data (such as frequency) has changed.

Using Delete Lists

Description

Delete Lists allow you to remove **non-content** bearing conjunctions, articles and other noise from texts. Delete List can be created internally in AutoMap or externally in a text editor. The list itself is a text file that contains a list (one concept per line) of the words to be deleted from the text.

NOTE : Whether you apply the Delete List(s) before or after applying a Thesauri will depend on your exact circumstances.

Step 1. Create a Delete List

There are two ways to create a new delete list:

Within AutoMap

Use the Concept List Viewer by select ***Tools => Concept List Viewer***. Place a check mark next to the concepts to include. Form the view menu select ***File => Save as Delete List***. The Delete List created can be viewed in the Delete List Editor by selecting ***Tools => Delete List Editor***.

Outside of AutoMap

Using a text editor or spreadsheet program capable of saving output as.txt files to manually create a Delete List. The main rule is one concept per line.

NOTE : Delete Lists can be opened in Excel, worked with, and then re-saved as a.txt file.

Step 2. Load Text Files

From the Pull Down Menu select ***File => Select Input Directory***. Navigate to a directory with your text files and click **Select**.

Step 3. Apply a Delete List

From the Pull Down menu select ***Preprocess => Apply Delete List***. Navigate to the file that contains your delete list and click **Select**.

Step 4. Select Type of Deletion

You will be prompted for the type of delete to perform. Direct will remove the concept entirely, whereas Rhetorical will replace the concept with xxx. Make your selection and click **OK**.

The Results

The results will appear in the Text Display Window.

Using a Generalization Thesaurus

Description

To use a unified key concept to represent many varieties of the same concept. For example to replace a contraction "don't" with its individual words "do not". This would be represented in the file as:

```
don't, do not
```

Be sure there are no extra spaces around the comma as they will be used in the translation. A spreadsheet program will not put in extra spaces.

Step 1. Review Your texts

Read through your texts to identify concepts to place into your thesaurus.

Step 2. Create a Thesaurus

You can create a thesaurus in either a text editor or a spreadsheet program that can save files as.csv files. The format of an entry is **concept,key_concept**. Concept can be single or multiple words and key_concept is one set of words usually separated by underscores.

```
US,United_States
United States,United_States
```

Step 3. Load Text Files

Place all your files in the same directory. Make sure that directory is empty before placing the files. From the Pull Down Menu select ***File => Select Input Directory***. Navigate to a directory with your thesaurus file and click **Select**.

Step 4. Apply Thesaurus

From the Pull Down Menu select ***Preprocess => Apply Generalization Thesauri***. Navigate to a directory with your thesauri and click **Select**. The results will be displayed in the Text Display Window.

Content Analysis to Semantic Network

Description

A semantic network will identify the relationships between concepts in the text.

Step 1. Load Text Files

Place all your files in the same directory. Make sure that directory is empty before placing the files. From the Pull Down Menu select ***File => Select Input Directory***. Navigate to a directory with your text files and click **Select**.

(Optional) Step 2. Create Concept Files

From the Pull Down Menu select ***Generate => Concept List***. Navigate to the directory to store these files (should be an empty directory) and click **Select**. AutoMap will ask if you want to create a Union Concept List. This will be useful for creating a Delete List on multiple files therefore click **Yes**.

(Optional) Step 3. Build a Generalization Thesauri

Review your texts for single concepts under multiple instances. (e.g., U.S. and United States can both be turned into United_States). In a text editor create an csv file with a list of entries consisting of a concept (one or more words in a file) and the new concept (all one string of words usually connected with an underscore) separated by a comma (e.g. U.S.,United_States and United States,United_States).

After constructing this file save it to a directory.

(Optional) Step 4. Apply a Generalization Thesauri

From the Pull Down Menu select ***Preprocess => Apply Generalization Thesauri***. Navigate to the directory containing your new thesaurus file, select a thesaurus, and click **Select**.

(Optional) Step 5. Build a Delete List

Open the Union Concept List with ***Tools => Concept List Viewer***. Place a check mark next to each concept you want placed in the Delete List. From the Pull Down Menu select ***File => Save Delete List*** and navigate to where you want to save it.

(Optional) Step 6. Apply a Delete List

From the Pull down Menu select ***Preprocess => Apply Delete List***. Navigate to the directory containing your delete List, highlight the file, and click **Select**. The preprocessed files will display in the Text Display Window.

Adjacency

When applying a delete list AutoMap will inquire as to the type of adjacency to use. The **Adjacency option** determines whether AutoMap will replace deleted concepts with a placeholder or not.

- **Direct Adjacency :** Removes concepts in the text that match concepts specified in the delete list and causes the remaining concepts to become adjacent.
- **Rhetorical Adjacency :** Removes concepts in the text that match concepts specified in the delete list and replaces them with **(xxx)**. The placeholders retain the original distances of the deleted concepts. This is helpful for visual analysis.

The newly pre-processed texts can be viewed in the main window.

Step 7. Create a Semantic Network

From the Pull Down Menu select ***Generate => Semantic Network***. AutoMap will generate one XML file for each text

loaded for use in ORA. Navigate to the directory to save these files and click Select.

AutoMap will output one XML file for each text file loaded. AutoMap will ask a couple of questions as to how you want to format the DyNetML file. You will be asked to select **Directionality** (Unidirectional or Bidirectional), **Window Size** (maximum distance between two concepts to be connected), **Stop Unit** (Clause, word, sentence, or paragraph), and **Number of [Stop Units]**.

Step 8. Load the DyNetML files in *ORA

Start *ORA and load the newly created **XML** files *ORA.

Multiple Delete Lists and Thesauri

Multiple delete lists and thesauri can be applied to the same text by loading, and applying the first delete list then loading, and applying a subsequent delete list. Any number can be applied in this manner. They can be viewed in order using the Pull Down Menu in the menu bar.

Un-apply a Delete List or a Thesaurus

Delete Lists and Thesauri can be **unapplied** but only in the same order that all preprocessing has been applied. If other preprocessing steps have been taken then you must Undo those steps also.

Modifying a Delete List

After a Delete list is created you can modify it using the **Delete List Editor**. From the Pull Down menu select ***Tools => Delete List Editor***. From the Viewer's Pull Down Menu select File => Open File and navigate to the directory containing your Delete Lists. Place a check mark in the **Select to Remove** column for concepts to remove from the Delete List. Typing concepts into the textbox and clicking **[Add Word]** will add concepts to the Delete List. When you are finished select ***File => Save as Delete List***.

Save text(s) after Delete List

You can save your texts after applying a delete list by selecting from the Pull Down Menu ***File => Save Preprocess Files***. This must be done before any other further preprocessing is performed as this option saves the texts at the highest level of preprocessing.

Interface Details

The Pull Down Menu

File

File => Select Input Directory loads all text files into AutoMap from the directory chosen. All.txt files in the directory will be loaded.

File => Import Text is similar to Select Input Directory as it loads all.txt files from one directory but provides additional support to load text files in other encodings. The default is **Let AutoMap Detect**.

File => Save Preprocessed Text Files saves all your files based on the highest level of preprocessing.

File => Exit will exit the AutoMap GUI program.

Edit

Edit => Set Font allows the user to change the font of the **Display Window**. The importance of changing the font is to display foreign character text. The font choices are based on the fonts available on the computer.

Preprocess

These options permit the cleaning and modification of the text in preparation of generating output. Contains the following preprocessing options: **Remove Extra Spaces, Remove Punctuation, Remove Symbols, Remove Numbers, Convert to Lowercase, Convert to Uppercase, Apply Stemming, Apply Delete List, & Apply Generalization Thesauri**.

These functions alter the text. They may be applied in any order as there should be no side effects.

Generate

Used for the generation of output from preprocessed files. The following output are available: **Concept List, Semantic List, Parts of Speech Tagging, Semantic Network, DyNetML MetaNetawork, Bigrams, Text Properties, Named entities, Feature Selection, Suggested MetaNetwork Thesauri, Union Concept Lists**.

These functions output files and are based on the highest level of preprocessing done.

Tools

AutoMap contains a number of Editors and Viewers for the files. These include: **Delete List Editor, Thesauri Editor, Concept List Viewer, Semantic List Viewer, DyNetML Network Viewer**.

These allow the user to edit support files used in preprocessing, or to view the results that have been generated.

Help

The Help file and about AutoMap.

Quick Launch Buttons

These buttons correspond to the functions in the Preprocess Menu.

File Navigation Buttons

Used to display the files in the main window. The buttons contain from left to right: **First, Previous, Goto, Next, and Last**.

Preprocess Order Window

Contains a running list of the preprocesses performed on the files. This can be undone one process at a time with the Undo command. The Undo affects the latest preprocess only.

Filename Box

Displays the name of the currently active file. Using the File Navigation Buttons will change this and as well as the text displayed in the window.

Text Display Window

Display the text for the file currently listed in the Filename Box.

Message Window

Area where AutoMap display the actions taken as well errors encountered.

01 JUL 09

Script Quickstart

The AM3Script is a command line utility that processes large numbers of files using a set of processing instructions provided in the configuration file. Some of the ways in which AutoMap is used:

- To extract a metanetwork representation of a dynamic/social network as expressed in text.
- To extract a semantic network to understand the relationships between concepts in texts.
- To clean and process text files for example by removing symbols and numbers, deleting unnecessary words, and stemming.
- To identify concepts and the frequency of concepts appearing in texts.

Description

AM3Script uses tags to tell AutoMap which functions to access. Functions are performed in the order they are listed in the config file. All preprocessing functions are followed by all processing functions and finally all post-processing functions are performed. Necessary output files are also written depending on the tags used in the config file.

If working with large numbers of texts it is best to use the script version as opposed to the GUI. The same processing steps available in the AutoMap GUI are available in the AutoMap Script.

Guide Roadmap

A. Script Overview

B. Tag List

C. Tutorial 1: Setting up a run in the Script

D. Tutorial 2: Using Delete Lists

E. Tutorial 3: Using a Thesauri

Before You Begin

AutoMap is a system that starts with text files. Before being able to use the features of AutoMap, it is necessary to have text to process. This text can be obtained from email, news articles, publications, web pages, or text typed in using a text editor.

AM3Script will process all text (.txt) files in a directory. It is not necessary to combine text into a single file. Some larger text files can be split into smaller text files to do analysis of sections individually.

It is suggested the user create sub-directories for input files, output, and support files all within an project directory. This assists in finding the correct files later and prevents AutoMap from overwriting previous files.

```
C:\My Documents\dave\project\input
C:\My Documents\dave\project\output
```

```
C:\My Documents\dave\project\support
```

Be sure to create the correct pathway in your config files to assure your files are written into the correct directory.

Running AutoMap Script

Once the configuration file has been created, the AM3Script is ready to use. The following is a brief on running the script.

1. Create a new .aos file. Configure the AM3Script .aos file as necessary by selecting the tags to use (Tag explanations in next section). Be sure to include pathways to input and output directories. Be sure to name the config file something unique.

   ```
   <Settings>
   <AutoMap
   textDirectory="C:\My Documents\dave\project\input"
   tempWorkspace="C:\My
   Documents\dave\project\output"
   textEncoding="unicode"/>
   </Settings>
   ```

2. Open a Command Prompt Window

3. Navigate to where the AutoMap3 program is installed. Mine is in Program Files. Yours could be in a different location.

   ```
   e.g. cd C:\Program Files\AM3
   ```

4. To run AM3Script type the following at the command prompt:

   ```
   am3script project.aos
   ```

   NOTE : project.aos is the name of my config file. Substitute the name of your config file. Also make sure there is a space between am3script and the name of your file.

5. AM3Script will execute using the .aos file specified.

   For Advanced Users

   It is possible to set the your PATH environmental variable to include the location of the install directory so that AM3Script can be used in any directory from the command

line. Please note this is not recommended for users that have no experience modifying the PATH environmental variable.

Script name

The script.aos file can be named whatever you like but we do recommend keeping the **.aos** suffix. This way you can do multiple runs to the files in a concise order: **step1.aos, step2.aos, step3.aos.**

Pathways (relative and absolute)

AM3Script config files allow you to specify pathways as either relative or absolute. It's important to know the difference. For relative pathways AutoMap always starts at the location of the AM3Script file. You can go up a directory with (..\) or down into a directory (\aDirectory). The last parameter will be the filename to use.

AM3Script resides in the directory where AutoMap was installed.

The pathway **..\input\aTextFile.txt** tells AutoMap to **go up one directory** then down into the **input** directory and find the file **aTextFile.txt**.

The pathway **C:\My Documents\dave\input\aTextFile.txt** tells AutoMap to start at the root directory of the hard drive and follow the designated pathway to the file.

NOTE : If given a non-existent pathway you will receive an error message during the run.

Tag Syntax in AM3Script

There are two styles of tags in the AM3Script. The first one uses a set of two tags. The first tag starts a section and the second tag ends the section. The second tag will contain the exact same word as the first but will have, in addition, a "/" appended after the word and before the ending bracket. This designates it as an ending tag. All the

parameters/attributes pertaining to this tag will be set-up between these two tags. e.g. **<aTag></aTag>**.

The second style is the self-ending tag as it contains a "/" within the tag. Any attributes used with this tag are contained within the tag e.g. **<aTag attribute="attributeName"/>**.

Output Directory syntax (TempWorkspace)

Output directories created within functions under the <PreProcessing> tag will all be suffixed with a number designating the order they were performed in. If a function is performed twice, each will have a separate suffix e.g. Generalization_3 and Generalization_5 denotes a Generalization Thesauri was applied to the text in the 3rd and 5th steps. Using thesauriLocation different thesauri could be used in each instance. For all other functions outside PreProcessing there is no suffix attached.

NOTE : The output directories specified above are in a temporary workspace and the content will be deleted if AM3Script uses this directory again in processing. It is recommended that the directory specified in the temp workspace be an empty directory. Also, for output that user wishes to keep from processing it is recommended to use the outputDirectory parameter within the individual processing step.

Example

```
<AddAttributes3Col attributeFile="C:\My
Documents\dave\project\attributeFiles\attribute
s.txt" outputDirectory="C:\My
Documents\dave\project\3ColAttribute\" />
```

By using these tags the user can specify where they want the individual processing step output to go. It also makes finding the location of the output files much simpler instead of looking through the contents of the TempWorkspace.

AM3Script Tags Details

<Script></Script> (required)

This set of tags is used to enclose the entire script. Everything used by the script must fall between these two tags. The only line found outside these tags will be the declaration line for xml version and text-encoding information: `<?xml version="1.0" encoding="UTF-8"?>`

<Settings></Settings> (required)

Used for the setting for the default directories for text and workspace. For AM3Script the tag is <AutoMap/>

NOTE : Any of the parameters can use inputDirectory and outputDirectory to override the default file location. These pathways will be relative to the location of the AM3Script.

<AutoMap/> (Required)

The <AutoMap/> tag contains default pathways used by all functions and the type of text encoding to use. Any function can override these pathways by setting inputDirectory and outputDirectory within it's own tag. The location of text files to process is contained in `textDirectory="C:\My Documents\dave\project\input"`. The location of the files that will be written to the output directory is in **class="sometext">tempWorkspace="C:\My Documents\dave\project\output"**. To specify the encoding method to use set **textEncoding="unicode"** (currently UTF-8 is the default. AutoMap uses UTF-8 for processing. Please make sure to set text encoding to your correct specification of your text.). **AutoDetect** will attempt to detect and convert your text over to UTF-8.

<Utilities></Utilities> (required)

The <Utilities> tag contains the sections <PreProcessing>, <Processing>, and <PostProcessing>. All three sections need to be nested within the <Utilities> tag in that order.

AutoMap 3 Preprocessing Tags

<PreProcessing></PreProcessing> (required)

These are utilities that modify raw text. The order the steps are placed in the file is the order they are performed.

You can also perform any of these utilities multiple times. e.g. perform a <Generalization/>, then a <DeleteList/>, then another <Generalization/>. Each step's results will be written to a separate output directory.

<RemoveNumbers/>

This parameter accepts either **whiteOut="y"** or **whiteOut="n"**. A "y" replaces numbers with spaces i.e. C3PO => C PO. A "no" removes the numbers entirely and closes up the remaining text e.g. C3PO => CPO.

```
<Script>
<Settings>
<AutoMap
textDirectory="C:\My Documents\dave\project\input"
tempWorkspace="C:\My Documents\dave\project\output"
textEncoding="unicode"/>
</Settings>
<Utilities>
<PreProcessing>
<RemoveNumbers whiteOut="y"/>
</PreProcessing>
<Processing>
</Processing>
<PostProcessing>
</PostProcessing>
</Utilities>
</Script>
```

<RemoveSymbols/>

This parameter accepts either **whiteOut="y"** or **whiteOut="n"**. A "y" replaces symbols with spaces. A "no" removes the symbols entirely and closes up the remaining text. The list of symbols that are removed: `~`@#$%^&*_+={}[]\|/<>`.

```
<Script>
<Settings>
<AutoMap
textDirectory="C:\My Documents\dave\project\input"
tempWorkspace="C:\My Documents\dave\project\output"
textEncoding="unicode"/>
</Settings>
<Utilities>
<PreProcessing>
<RemoveSymbols whiteOut="y"/>
</PreProcessing>
```

```
<Processing>
</Processing>
<PostProcessing>
</PostProcessing>
</Utilities>
</Script>
```

<RemovePunctuation/>

This parameter accepts either **whiteOut="y"** or **whiteOut="n"**. A "y" replaces punctuation with spaces. A "no" removes the punctuation entirely and closes up the remaining text. The list of punctuation removed is: `.,:;' "()!?-`.

```
<Script>
<Settings>
<AutoMap
textDirectory="C:\My Documents\dave\project\input"
tempWorkspace="C:\My Documents\dave\project\output"
textEncoding="unicode"/>
</Settings>
<Utilities>
<PreProcessing>
RemovePunctuation whiteOut="y"/>
</PreProcessing>
<Processing>
</Processing>
<PostProcessing>
</PostProcessing>
</Utilities>
</Script>
```

<RemoveExtraWhiteSpace/>

Find instances of multiple spaces and replaces them a single space. Note, there are no extra parameters for this step. It's only function is to reduce multiple spaces to one space.

```
<Script>
<Settings>
<AutoMap
textDirectory="C:\My Documents\dave\project\input"
tempWorkspace="C:\My Documents\dave\project\output"
textEncoding="unicode"/>
</Settings>
<Utilities>
<PreProcessing>
RemoveExtraWhiteSpace />
```

```
</PreProcessing>
<Processing>
</Processing>
<PostProcessing>
</PostProcessing>
</Utilities>
</Script>
```

<Generalization/>

The Generalization Thesaurus are used to replace possibly confusing concepts with a more standard form. e.g. a text contains both `United States` and `U.S.` The Generalization Thesaurus could have two entries which replace both the original entries with `united_states`.

If **useThesauriContentOnly="n"** AutoMap replaces concepts in the Generalization Thesaurus but leaves all other concepts intact. If **useThesauriContentOnly="y"** then AutoMap replaces concepts but removes all concepts not found in the thesaurus.

The other parameter is **thesauriLocation**. This allows you to specify the pathway to the thesaurus file to use.

The questions now is whether to use one big thesaurus or several smaller thesauri. When trying to replicate results over many runs using one file is easier to replicate.

The order of the thesauri entries will skew the results. (e.g. if you have both John & John Smith you need to put John Smith first. If John is listed first the end result will be John_Smith_Smith.

```
<Script>
<Settings>
<AutoMap
textDirectory="C:\My Documents\dave\project\input"
tempWorkspace="C:\My Documents\dave\project\output"
textEncoding="unicode"/>
</Settings>
<Utilities>
<PreProcessing>
Generalization thesauriLocation="C:\My
Documents\dave\project\support\thesauri.csv"
useThesauriContentOnly="y" />
</PreProcessing>
<Processing>
```

```
</Processing>
<PostProcessing>
</PostProcessing>
</Utilities>
</Script>
```

<DeleteList/>

The Delete List is a list of concepts (one concept per line) to remove from the text files before output file. Set **adjacency="d"**, for direct (removes the space left by deleted words) and remaining concepts now become "adjacent" to each other. Set **adjacency="r"** for rhetorical (removes the concepts but inserts a spacer within the text to maintain the original distance between concepts).

The other parameter is **deleteListLocation** which specifies the pathway to the Delete List.

```
<Script>
<Settings>
<AutoMap
textDirectory="C:\My Documents\dave\project\input"
tempWorkspace="C:\My Documents\dave\project\output"
textEncoding="unicode"/>
</Settings>
<Utilities>
<PreProcessing>
DeleteList adjacency="r" deleteListLocation="C:\My
Documents\dave\project\support\deleteList.txt" saveTexts="y"/>
</PreProcessing>
<Processing>
</Processing>
<PostProcessing>
</PostProcessing>
</Utilities>
</Script>
```

<FormatCase/>

FormatCase changes the output text to either "lower" or "upper" case. If **changeCase="l"** then AutoMap will change all text to lowercase. **changeCase="u"** changes nall text to uppercase.

```
<Script>
<Settings>
<AutoMap
```

```
textDirectory="C:\My Documents\dave\project\input"
tempWorkspace="C:\My Documents\dave\project\output"
textEncoding="unicode"/>
</Settings>
<Utilities>
<PreProcessing>
<FormatCase changeCase="u"/>
</PreProcessing>
<Processing>
</Processing>
<PostProcessing>
</PostProcessing>
</Utilities>
</Script>
```

<Stemming/>

Stemming removes suffixes from words. This assists in counting similar concepts in the singular and plural forms (e.g. plane and planes). These concepts would normally be considered two terms. After stemming planes becomes plane and the two concepts are counted together.

There are two stemming options: **type="k"** uses the KSTEM or Krovetz stemmer and **type="p"** uses the Porter stemmer.

```
<Script>
<Settings>
<AutoMap
textDirectory="C:\My Documents\dave\project\input"
tempWorkspace="C:\My Documents\dave\project\output"
textEncoding="unicode"/>
</Settings>
<Utilities>
<PreProcessing>
<Stemming type="k" />
</PreProcessing>
<Processing>
</Processing>
<PostProcessing>
</PostProcessing>
</Utilities>
</Script>
```

<Processing> (required)

These steps are performed after all "Pre-Processing" is finished. They are performed in the order they appear in the AM3Script.

<POSExtraction/>

posType="ptb" specifies a tag for each part of speech. **posType="aggregate"** groups many categories together using fewer Parts-of-Speech tags.

```
<Script>
<Settings>
<AutoMap
textDirectory="C:\My Documents\dave\project\input"
tempWorkspace="C:\My Documents\dave\project\output"
textEncoding="unicode"/>
</Settings>
<Utilities>
<PreProcessing>
<Processing>
<posType="ptb" />
</PreProcessing>
<Processing>
</Processing>
<PostProcessing>
</PostProcessing>
</Utilities>
</Script>
```

<Anaphora />

An anaphoric expression is one represented by some kind of deictic, a process whereby words or expressions rely absolutely on context. Sometimes this context needs to be identified. These definitions need to be specified by the user. Used primarily for finding personal pronouns, determining who it refers to, and replacing the pronoun with the name.

```
<Script>
<Settings>
<AutoMap
textDirectory="C:\My Documents\dave\project\input"
tempWorkspace="C:\My Documents\dave\project\output"
textEncoding="unicode"/>
</Settings>
<Utilities>
<PreProcessing>
<Processing>
```

```
<posType="ptb" />
</PreProcessing>
<Processing>
</Processing>
<PostProcessing>
</PostProcessing>
</Utilities>
</Script>
```

NOTE : For Anaphora to work POS must be run first.

<ConceptList />

Creates a separate list of concepts for each loaded text file. A Delete List or Generalization Thesauri can be performed before creating these lists to reduce the number of concepts needed to be included in this file. These concept Lists can be loaded into a spreadsheet and sorted by any of the headers.

```
<Script>
<Settings>
<AutoMap
textDirectory="C:\My Documents\dave\project\input"
tempWorkspace="C:\My Documents\dave\project\output"
textEncoding="unicode"/>
</Settings>
<Utilities>
<PreProcessing>
<Processing>
<ConceptList />
</PreProcessing>
<Processing>
</Processing>
<PostProcessing>
</PostProcessing>
</Utilities>
</Script>
```

<SemanticNetwork/>

A semantic network displays the connection between a text's concepts. These links are defined by four parameters. **windowSize:** the distance two concepts can be apart and have a relationship. textUnit defined as (S)entence, (W)ord, (C)lause, or (P)aragraph. resetNumber defines the number of textUnits to process before resetting the window. directional defined as Unidirectional (which looks forward only in the text file) or

Bi-Directional (which finds relationships in either direction).

```
<Script>
<Settings>
<AutoMap
textDirectory="C:\My Documents\dave\project\input"
tempWorkspace="C:\My Documents\dave\project\output"
textEncoding="unicode"/>
</Settings>
<Utilities>
<PreProcessing>
<Processing>
<SemanticNetwork windowSize="2 textUnit="S" resetNumber="2"
directional="U" />
</PreProcessing>
<Processing>
</Processing>
<PostProcessing>
</PostProcessing>
</Utilities>
</Script>
```

<MetaNetwork/>

This associates text-level concepts with Meta-Network categories (e.g. agent, resource, knowledge, location, event, group, task, organization, role, action, attributes, when). Concepts can be translated into multiple Meta-Network categories. thesauriLocation="C:\My Documents\dave\project\support" designates the location of the MetaNetwork Thesauri, when used.

```
<Script>
<Settings>
<AutoMap
textDirectory="C:\My Documents\dave\project\input"
tempWorkspace="C:\My Documents\dave\project\output"
textEncoding="unicode"/>
</Settings>
<Utilities>
<PreProcessing>
<Processing>
MetaNetwork thesauriLocation="C:\My
Documents\dave\project\support\thesauri.csv" />
</PreProcessing>
<Processing>
</Processing>
<PostProcessing>
</PostProcessing>
```

```
</Utilities>
</Script>
```

<UnionConceptList />

Union Concept Lists is a list of concepts taken from all texts currently loaded, rather than only one text file. It reports total frequency, related frequency, and cumulative frequencies of concepts in all text sets. It's helpful in finding frequently occurring concepts over all loaded texts.

```
<Script>
<Settings>
<AutoMap
textDirectory="C:\My Documents\dave\project\input"
tempWorkspace="C:\My Documents\dave\project\output"
textEncoding="unicode"/>
</Settings>
<Utilities>
<PreProcessing>
<Processing>
<UnionConceptList />
</PreProcessing>
<Processing>
</Processing>
<PostProcessing>
</PostProcessing>
</Utilities>
</Script>
```

NOTE : The number of unique concepts considers each concept only once, whereas the number of total concepts considers repetitions of concepts.

<NGramExtraction />

NGramExtraction creates a file listing all the NGrams, their frequency in the files, their relative frequency to each other, and the gram type.

```
<Script>
<Settings>
<AutoMap
textDirectory="C:\My Documents\dave\project\input"
tempWorkspace="C:\My Documents\dave\project\output"
textEncoding="unicode"/>
</Settings>
<Utilities>
<PreProcessing>
```

```
<Processing>
<NGramExtraction />
</PreProcessing>
<Processing>
</Processing>
<PostProcessing>
</PostProcessing>
</Utilities>
</Script>
```

<SemanticNetworkList />

Creates a file consisting of pairs of concepts and their frequency within the text files. This takes four parameters: windowSize: the distance two concepts can be apart and have a relationship. textUnit defined as (S)entence, (W)ord, (C)lause, or (P)aragraph. resetNumber defines the number of textUnits to process before resetting the window. directional defined as Unidirectional (which looks forward only in the text file) or Bi-Directional (which finds relationships in either direction).

```
<Script>
<Settings>
<AutoMap
textDirectory="C:\My Documents\dave\project\input"
tempWorkspace="C:\My Documents\dave\project\output"
textEncoding="unicode"/>
</Settings>
<Utilities>
<PreProcessing>
<Processing>
<SemanticNetworkList directional="U" resetNumber="1" textUnit="S"
windowsize="5" />
</PreProcessing>
<Processing>
</Processing>
<PostProcessing>
</PostProcessing>
</Utilities>
</Script>
```

<PostProcessing></PostProcessing> (required)

The PostProcessing section contains functions to perform after all Processing steps are complete.

<addAttributes>

Additional attributes can be added to the nodes within the generated DyNetML file. attributeFile="C:\My Documents\dave\project\support\attribute_file" is the pathway to the file which contains a header row with the attribute name.

```
<Script>
<Settings>
<AutoMap
textDirectory="C:\My Documents\dave\project\input"
tempWorkspace="C:\My Documents\dave\project\output"
textEncoding="unicode"/>
</Settings>
<Utilities>
<PreProcessing>
<Processing>
<addAttributes attributeFile="C:\My
Documents\dave\project\support\attribute_file" />
</PreProcessing>
<Processing>
</Processing>
<PostProcessing>
</PostProcessing>
</Utilities>
</Script>
```

This is similar to <addAttributes> but uses name and value.

```
<Script>
<Settings>
<AutoMap
textDirectory="C:\My Documents\dave\project\input"
tempWorkspace="C:\My Documents\dave\project\output"
textEncoding="unicode"/>
</Settings>
<Utilities>
<PreProcessing>
<Processing>
<addAttributes3Col attributeFile="C:\My
Documents\dave\project\support\3Col_file" />
</PreProcessing>
<Processing>
</Processing>
<PostProcessing>
</PostProcessing>
</Utilities>
</Script>
```

<UnionDynetml/>

UnionDynetml creates a union of all dynetml in a specified directory. It requires a unionType which is s or m. "s" is for a union of semantic networks and "m" is for metanetworks.

```
<Script>
<Settings>
<AutoMap
textDirectory="C:\My Documents\dave\project\input"
tempWorkspace="C:\My Documents\dave\project\output"
textEncoding="unicode"/>
</Settings>
<Utilities>
<PreProcessing>
<Processing>
<UnionDynetml unionType="s" />
</PreProcessing>
<Processing>
</Processing>
<PostProcessing>
</PostProcessing>
</Utilities>
</Script>
```

Simple Tutorials

Setting Up and Using Thesauri

Description

Thesauri are used to reduce the number of unique concepts in the texts by assigning a key concept to multiple versions of the same concept. This example uses the file structure below. Your file strauctur may differ

```
C:\My Documents\dave\project\input
C:\My Documents\dave\project\output
C:\My Documents\dave\project\support
```

Step 1: Examining the text

If you know the subject matter then many of the multiple versions of a concept will be known already. Other times it will be necessary to examine the text to determine what those concepts are.

```
Ted is a U.S. citizen. He lives in the United
States. Ted says, I love living in America.
```

There are three concepts that all mean the same thing:
U.S., the United States, and America.

Step 2: Creating a Thesauri

Once the multiple word concepts are identified you can create a thesauri to combine them into key concepts. Remember:

1. One concept per line
2. Concept and key concept separated by a comma (no spaces before or after the comma)
3. Concept can be multiple words
4. Key concept can only be one word but may contain dividing punctuation (underscores are mainly used for this purpose.

```
U.S.,the_United_States_of_America
the United States,the_United_States_of_America
America,the_United_States_of_America
```

Save this file as a .csv file.

Step 3: Using in the .aos file

Place the tag <Generalization thesauriLocation="C:\My Documents\dave\support\genThes.csv" useThesauriContentOnly="y"> in the <PreProcessing> section. Select whether to use thesauri content only: y (make thesaurus replacements but output only the concepts listed in the thesaurus) or n (no: make thesaurus replacements but output all concepts). Place the pathway to your newly created Thesaurus in the thesauriLocation parameter.

Step 4: Run the script

Open a Command Run window and navigate to the directory where AutoMap3 was installed. At the prompt type am3script project.aos file. When finished navigate to the output directory denoted in the.aos file to find your output files.

Step 5: View the Results

Open the newly created text files in a text editor to review.

Setting Up and Using Delete Lists

Description

Delete Lists can be created using a text editor or spreadsheet program.

Step 1: Creating Delete Lists with a text editor

Open your text editor or spreadsheet and create a list of concepts to use as a Delete List. Place only one, single word concept per line. Save as a .txt file.

Step 2. Make a new .aos file

Make a copy of the standard .aos file and open it in a text editor. Specify where your input are and ahere to write the output files

```
textDirectory="C:\My Documents\dave\project\input"
tempWorkspace="C:\My
Documents\dave\project\output"
```

Save this new .aos file in the same directory as the AM3Script file. If the file is not in the same directory then AM3Script will fail.

Step 3: Using in the .aos file

Place the tag <DeleteList adjacency="" deleteListLocation=""> in the <PreProcessing> section. For adjacency select d (direct: totally remove deleted concepts) or r (rhetorical: replace deleted concepts with a placeholder). Place the pathway to your newly created Delete List in the deleteListLocation parameter.

```
<DeleteList adjacency="r"
deleteListLocation="C:\My
Documents\dave\project\support\deleteList.txt"
/>
```

Step 4: Run the script

Navigate to the directory containing AutoMap3. At the command prompt type am3script project.aos. Check for your output files in the directory designated in the .aos file

Setting up and Running a Script

Description

Running the script requires the use of the Command Line Prompt. This is found in the Start menu. It's exact location may be different depending on the setup of your particular computer. It is normally found in the "All Programs" option in the "Accessories" directory.

Step 1. Create WorkSpace for input & output files

Navigate to your workspace and create a project directory. Inside this directory create and input and output directory.

```
C:\My Documents\dave\project
C:\My Documents\dave\project\input
C:\My Documents\dave\project\output
C:\My Documents\dave\project\support
```

Step 2. Place your text files in the input directory

Copy all your text files into the C:\My Documents\dave\project\input directory.

Step 4. Place your work files in a directory

Place any Delete Lists and Thesauri in the C:\My Documents\dave\project\support directory.

Step 5. Make a new .aos file

Make a copy of the standard .aos file and open it in a text editor. Tell AutoMap where your input files are and where you want the output written. Under the <AutoMap> tag is textDirectory (where you placed your text files) and tempWorkSpace (where you want AutoMap to write your output files. This config file is setup to apply a thesarus, apply a delete list, and produce concept lists.

Step 6. Determine the Preprocessing functions to use

Review the list of AutoMap tags to determine which script tags will be necessary. Insert those tags into your new .aos file in the proper location. Set the parameters for each function you are using. Some functions do not require any additional parameters while others require to tell AutoMap the type of processing you want.

```
<?xml version="1.0" encoding="UTF-8"?>
<Script>
<Settings>
<AutoMap
textDirectory="C:\My Documents\dave\project\input"
tempWorkspace="C:\My
Documents\dave\project\output"
textEncoding=""/>
</Settings>
<Utilities>

<PreProcessing>
<Generalization thesauriLocation="C:\My
Documents\dave\project\thesauri.csv"
useThesauriContentOnly="y" />
<DeleteList adjacency="r"
deleteListLocation="C:\My
Documents\dave\project\deleteList.txt"
saveTexts="y"/>
</PreProcessing>

<Processing>
<ConceptList />
<UnionConceptList />
</Processing>

<PostProcessing>
</PostProcessing>

</Utilities>
</Script>
```

Save this new .aos file in the same directory as the AM3Script file. If the file is not in the same directory then AM3 Script will fail.

Step 7. Open a Command Window

From the Start Menu open a Command Run Window. By default this is in the Accessories folder but may be in a different location on your machine. Navigate to the location which contains AM3Script.

Step 8. Run the Script file

Navigate to the directory containing AutoMap. At the prompt type am3script project.aos

11 NOV 10

Non-English Fonts

Description

Many languages use non-Latin fonts with characters not found in the standard set. Latin text sets use a single byte character set. Asian sets (like Chinese) use a double-byte text set. Many fonts require a different method of installation. We suggest you refer to your manual for the proper way to install new fonts for your particular computer and/or operating system.

How can you tell if you need to download a font? Sometimes the fonts are already available on your computer and it's just a matter of changing the settings on your computer so that you can access them. Typically, the newer the operating system, the more languages it will support straight out of the box.

Font Web Sites

The following web sites contain fonts for various non-English languages.

Vistawide World Languages and Culture : A collection of free non-English fonts and information on activating them on your computer.

http://www.vistawide.com/languages/foreign_language_fonts.htm

TypeNow : A collection of free non-English fonts in zip format.

http://www.typenow.net/language.htm

kwintessential : A collection of free non-English fonts.

http://www.kwintessential.co.uk/fonts/foreign-language.html

freelang.net : A collection of free non-English fonts for Windows.

http://www.freelang.net/fonts/index.php

Installation

Guide to Installing East Asian Languages : This page outlines the steps for installing East Asian languages on a computer running Windows. There are pages for Windows 2000 Pro, XP.

http://newton.uor.edu/Departments&Programs/AsianStudiesDept/Language/index.html

Pinyin Joe's Chinese Computing Help Desk : contains information on activating Chinese fonts in XP.

http://www.pinyinjoe.com/pinyin/pinyin_XPfonts.htm

Resources

South Asia Language Resource : The South Asia Language Resource Center is a collaborative effort funded by a grant from the U.S. Department of Education's International Education and Graduate Programs Service. The Language Resource Center at the University of Chicago is one of fifteen nationwide that exist to improve the capacity to teach and learn foreign languages effectively. SALRC primarily focuses on the needs concerning South Asian language pedagogy in American universities.

http://salrc.uchicago.edu/resources/fonts/available/urdu/

23 OCT 09

Java Licenses

Description

This table contains all the libraries used in AutoMap 3 along with the web sites for download and licenses.

Library	Name / Website / License
activem1-all-5.4.0.jar	Apache ActiveMQ http://commons.apache.org Apache License Version 2.0
abdera-core-1.0.jar	Apache Abdera http:/abdera.apache.org/ The Apache Software License Version 2.0
abdera-extensions-json-1.0.jar	Apache Abdera http:/abdera.apache.org/ The Apache Software License Version 2.0
abdera-extensions-main-1.0.jar	Apache Abdera http:/abdera.apache.org/ The Apache Software License Version 2.0
abdera-i18n-1.0.jar	Apache Abdera http:/abdera.apache.org/ The Apache Software License Version 2.0
abdera-parser-1.0.jar	Apache Abdera http:/abdera.apache.org/ The Apache Software License Version 2.0
ant-1.6.5.jar	Apache Ant http:/ant.apache.org/ The Apache Software License Version 2.0
antlr-2.7.7.jar	ANTLR http:www.antlr.org/license The Apache Software License Version 2.0
aopalliance-1.0.jar	AOP Alliance All the source code provided by AOP Alliance is Public Domain.
asm-2.2.3.jar	ASM http://asm.ow2.org/license.html

axiom-api-1.2.8_1.jar	Apache Axiom http:/ws.apache.org/axiom Apache License 2.0
axiom-impl-1.2.7.jar	Apache Axiom http:/ws.apache.org/axiom Apache License 2.0
axiom.jar	Apache Axiom xml Object Model http://axiom.apache.org/commons/axiom/index.html The Apache Software License, Version 2.0M
bcprov-jdk15-1.43.jar	Bouncy Castle http://www.bouncycastle.org/licence.html
colt.jar	Colt Project http://acs.lbl.gov/software/colt/ http://acs.lbl.gov/software/colt/license.html
commons-codec-1.3.jar	Apache Commons Codec http://commons.apache.org/codec/ The Apache Software License, Version 2.0
commons-httpclient-3.1.jar	Apache Commons http://commons.apache.org/ The Apache Software License, Version 2.0
commons-lang-2.4.jar	http://commons.apache.org/ Apache License 2.0
commons-logging-1.1.1.jar	Apache Commons Logging http://commons.apache.org/logging/ The Apache Software License, Version 2.0
commons-logging-adapters-1.1.jar	Apache Commons Logging Adapters http://commons.apache.org/logging/ The Apache Software License, Version 2.0
commons-logging-api-1.1.1.jar	Apache Commons Logging API http://commons.apache.org/logging/ The Apache Software License, Version 2.0

commons-net-2.0.jar	Apache Commons Net http://commons.apache.org/net/ The Apache Software License, Version 2.0
commons-pool-1.5.2.jar	Apache Commons http://commons.apache.org The Apache Software License, Version 2.0
crf.jar	Conditional Random Fields http://crf.sourceforge.net/ Sunita Sarawagi of IIT Bombay
cxf-2.2.9.jar	Apache CXF http://commons.apache.org Apache License Version 2.0
cxf-manifest.jar	Apache CXF http://commons.apache.org Apache License Version 2.0
cxf-xjc-boolean-2.2.9.jar	Apache CXF http://commons.apache.org Apache License Version 2.0
cxf-xjc-bug671-2.2.9.jar	Apache CXF http://commons.apache.org Apache License Version 2.0
cxf-xjc-dv-2.2.9.jar	Apache CXF http://commons.apache.org Apache License Version 2.0
cxf-xjc-ts-2.2.9.jar	Apache CXF http://commons.apache.org Apache License Version 2.0
FastInfoset-1.2.7.jar	Fast Infoset Project http://commons.apache.org Apache License Version 2.0
google-collect-1.0-rc1.jar	Guava http://code.google.com/p/guava-libraries/ The Apache Software License, Version 2.0
geromnimo-javamail_1	Apache Geronimo http://geronimo.apache.org/ The Apache Software License, Version 2.0

.4_spec-1.6.jar	
geronimo-activation_1.1_spec-1.0.2.jar	Apache Geronimo http://geronimo.apache.org/ Apache License 2.0
geronimo-annotation_1.0_spec-1.1.1.jar	Apache Geronimo http://geronimo.apache.org/ Apache License 2.0
geronimo-jaxws_2.1_spec-1.0.jar	Apache Geronimo http://geronimo.apache.org/ Apache License 2.0
geromnimo-jms_1.1_spec_1.1.1.jar	Apache ActiveMQ http://geronimo.apache.org/ The Apache Software License, Version 2.0
geronimo-servlet_2.5_spec-1.2.jar	Apache Geronimo http://geronimo.apache.org/ Apache Geronimo
geronimo-stax-api_1.0_spec-1.0.1.jar	Apache Geronimo http://geronimo.apache.org/ Apache License 2.0
geronimo-ws-metadata_2.0_spec-1.1.2.jar	Apache Geronimo http://geronimo.apache.org/ Apache License 2.0
hibernate-core-3.3.2.GA.jar	Hibernate Core LGPL v2.1

htmlparser.jar	HTML Parser http://htmlparser.sourceforge.net/ Common Public License 1.0, GNU Library or Lesser General Public License (LGPL)
httpclient-4.0-beta2.jar	Jakarta Commons HTTP Client http://hc.apache.org/httpclient-3.x/ The Apache Software License, Version 2.0
httpcore-4.0-beta3.jar	Jakarta Commons HTTP Client http://hc.apache.org/httpclient-3.x/ The Apache Software License, Version 2.0
httpcore-nio-4.0-beta3.jar	Jakarta Commons HTTP Client http://hc.apache.org/httpclient-3.x/ The Apache Software License, Version 2.0
httpmime-4.0-beta2.jar	Jakarta Commons HTTP Client http://hc.apache.org/httpclient-3.x/ The Apache Software License, Version 2.0
itext-2.1.6.jar	iText PDF http://itextpdf.com/ Affero General Public License (AGPL)
jaxb-api-2.1.jar	Oracle JAXB http://download.oracle.com/docs/cd/E17802_01/webservices/webservices/docs/1.5/jaxb/index.html
jaxb-impl-2.1.13.jar	Oracle JAXB http://download.oracle.com/docs/cd/E17802_01/webservices/webservices/docs/1.5/jaxb/index.html
jaxb-xjc-2.1.13.jar	Oracle JAXB http://download.oracle.com/docs/cd/E17802_01/webservices/webservices/docs/1.5/jaxb/index.html
jaxen-1.1.1.jar	Jaxen http://jaxen.codehaus.org/license.html
jdom-1.1.jar	JDOM http://www.jdom.org/ JDOM is available under an Apache-style open source license, with the acknowledgment clause

	removed. This license is among the least restrictive license available, enabling developers to use JDOM in creating new products without requiring them to release their own products as open source. This is the license model used by the Apache Project, which created the Apache server. The license is available at the top of every source file and in LICENSE.txt in the root of the distribution.
jettison-1.2.jar	**Jettison Apache License 2.0**
jetty-6.1.21.jar	**Jetty http://www.eclipse.org/jetty/licenses.php**
jetty-util-6.1.21.jar	**Jetty http://www.eclipse.org/jetty/licenses.php**
jide-oss-2.8.4.jar	**JIDE Common Layer Open Source Project https://jide-oss.dev.java.net/ JIDE Common Layer is dual-licensed. The two licenses are GPL with classpath exception and free commercial license.**
jra-1.0-alpha-4.jar	**Java REST Annotations Apache License 2.0**
js-1.7R1.jar	**Mozilla Rhino Mozilla Public license version 1.1**
jsoup-1.5.2.jar	**Remove HTML tags https://jsoup.org/license The MIT License**
json-20070829.jar	**JSON http://www.json.org/java/index.html http://www.JSON.org/license.html**
jsr311-api-1.0.jar	**JSR 311 CDDL License**
jta-1.1.jar	**Java TRansaction API http://www.oracle.com/technetwork/java/javaee/jta/index.html**
lbfgs.jar	**The RISO Project**

	http://riso.sourceforge.net/ http://www.gnu.org/copyleft/gpl.html
log4j-1.2.13.jar	Apache Log4j http://commons.apache.org Apache License Version 2.0
log4j-1.2.15	Apache log4j http://logging.apache.org/log4j/1.2/ The Apache Software License, Version 2.0
lucene-jar.jar	Apache Lucene http://lucene.apache.org/java/docs/ The Apache Software License, Version 2.0
mime-util-2.1.3.jar	MIME type detection utility The Apache Software License, Version 2.0
neethi-2.0.4.jar	Neethi http://commons.apache.org Apache License Version 2.0
nekohtml.jar	CyberNeko HTML Parser http://nekohtml.sourceforge.net/ The Apache Software License, Version 2.0
ode-tools.jar	Apache Ode http://ode.apache.org/ The Apache Software License, Version 2.0
ode-utils.jar	Apache Ode http://ode.apache.org/ The Apache Software License, Version 2.0
oro-2.0.8.jar	Apache Jakarta Project Oro http://commons.apache.org Apache License Version 2.0
pdftotextproject.jar	contains packages from http://pdfbox.apache.org/
poi-3.2-final-20081019.jar	Apache POI http://poi.apache.org/ The Apache Software License, Version 2.0

poi-scratchpad-3.2-final-20081019.jar	Apache POI http://poi.apache.org/ The Apache Software License, Version 2.0
rome-1.0.jar	Project ROME https://rome.dev.java.net/ The Apache Software License, Version 2.0
saaj-api-1.3.jar	SOAP with Attachments API Package COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0
saaj-impl-1.3.2.jar	SOAP with Attachments API Package COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0
serializer-2.7.1.jar	Xalan Java Serializer http://commons.apache.org Apache License Version 2.0
slf4j-api-1.5.8.jar	Simple Logging Facade for Java http://www.slf4j.org/license.html
slf4j-jdk14-1.5.8.jar	Simple Logging Facade for Java http://www.slf4j.org/license.html
spring-beans-2.5.6.jar	Spring Framework http://commons.apache.org Apache License Version 2.0
spring-context-2.5.6.jar	Spring Framework http://commons.apache.org Apache License Version 2.0
spring-context-support-2.5.6.jar	Spring Framework http://commons.apache.org Apache License Version 2.0
spring-core-2.5.6.jar	Spring Framework License: Apache License Version 2.0
spring-	Spring Framework

jms-2.5.6.jar	http://commons.apache.org Apache License Version 2.0
spring-tx-2.5.6.jar	Spring Framework http://commons.apache.org Apache License Version 2.0
spring-web-2.5.6.jar	Spring Framework http://commons.apache.org Apache License Version 2.0
sptoolkit.jar	Sentence and Paragraph Breaker http://text0.mib.man.ac.uk:8080/scottpiao/sent_detector Scott Piao, School of Computer Science, Manchester UNiversity, UK
velocity-1.6.4.jar	Apache Velocity http://commons.apache.org Apache License Version 2.0
websphinx.jar	Web Sphinx http://www.cs.cmu.edu/~rcm/websphinx/ The Apache Software License, Version 2.0
wsdl4j-1.6.2.jar	Web Services Description Lanugage for Java Toolkit Common Public License 1.0
wss4j-1.5.8.jar	Apache WSS4J http://commons.apache.org Apache License Version 2.0
wstx-asl-3.2.9.jar	Woodstox http://commons.apache.org Apache License Version 2.0
xalan-2.7.1.jar	Xalan Java http://commons.apache.org Apache License Version 2.0
xercesimpl-2.7.1.jar	Xerces2 Java Parser http://xerces.apache.org/xerces2-j/ The Apache Software License, Version 2.0
xml-apis.jar	Apache XML http://xml.apache.org/commons/

	The Apache Software License, Version 2.0
xml-apis-1.0.b2.jar	**XML Commons External Components XML APIs** **http://commons.apache.org** **Apache License Version 2.0**
xml-resolver-1.2.jar	**XML Commons Resolver Component** **http://commons.apache.org** **Apache License Version 2.0**
xmlbeans-2.4.0.jar	**xmlbeans-2.4.0.jar** **http://commons.apache.org** **Apache License Version 2.0**
XmlSchema-1.4.5.jar	**Apache XML Schema** **The Apache Software License, Version 2.0**
xmlsec-1.4.3.jar	**XML Security** **http://commons.apache.org** **Apache License Version 2.0**
xml-writer.jar	**XMLWriter** **http://www.megginson.com/downloads/**

Supplemental

Library	Name / Website / License
soracsc.jar	
soracsc-data.jar	
soracscregistry.jar	

11 AUG 11

Content Section

This section contains general explanations of the functions of AutoMap. It details the "What it is" aspect.

Anaphora
BiGrams
Concept Lists
Data Selection
Delete Lists
Encoding
File Formats
Format Case
MetaNetwork
Named Entity
Networks
Parts of Speech
Process Sequencing
Semantic Lists
Semantic Networks
Stemming
Text Properties
Thesauri, General
Thesauri, MetaNetwork
Thesaurus Content Only
Thresholds
Unions
Union Concept List
Window Size

Anaphora

Description

An anaphoric expression is one represented by some kind of deictic, a process whereby words or expressions rely absolutely on context. Sometimes this context needs to be identified. These definitions need to be specified by the user. Used primarily for finding personal pronouns, determining who it refers to, and replacing the pronoun with the name.

NOTE : Not all anaphora are pronouns and not all pronouns are anaphora.

Definition of Anaphora

Repetition of the same word or phrase at the start of successive clauses.

milkAndCookies.txt

```
Dave wants milk and cookies. He drives to the store.
He then buys milk and cookies.
```

The **He** at the beginning of the last two sentences are anaphoric under the strict definition (he refers to Dave).

What is NOT an anaphora

Not all pronouns are anaphoras. If there is no reference to a particular person then it remains justs a pronoun.

```
He who hesitates is lost.
```

The **He** at the beginning is NOT an anaphora as it does not refer to anyone in particular.

23 SEP 09

Bi-Grams

Description

BiGrams are two adjacent concepts in the same sentence. Two concepts are not considered a bigram if they are in separate sentences or paragraphs. If a Delete List is run previous to detecting bi-grams then the concepts in the Delete List are ignored. Multiple Delete Lists can be used with a set of files.

Definitions

Frequency:
the number of times that bi-gram occurs in a single text.

Relative Frequency:
The number of times a bi-gram occurs in a single text divided by the maximum occurrence of any bi-gram.

Maximum Occurrence:

The number of times that the bi-gram that occurred the most, occurred in a text.

Relative Percentage:
The percentage of all bi-grams accounted for by the occurrence of this bi-gram.

The Most Common BiGram

Not all bigrams are important. In fact, the most common bigram, **of the**, is usually very unimportant by itself.

For example, in the movie title **Lord of the Rings** the important words are **Lord** and **Rings**. But without the bigram **of the** the title would make no sense: **Lord Rings**. By itself **of the** has no meaning, but within another set of words helps create the proper context.

Changes in Meaning

When individual concepts are formed into bigrams their meanings can change.

Threshold in regards to BiGrams

Threshold is used to detect if there are specific number of occurrences of a Bi-Gram in the text(s). For **Global Threshold** a Bi-gram is detected if the total number of its occurrences in all texts is greater than or equal to the Global Threshold. For **Local Threshold** a Bi-gram is detected if the number of its occurrences in EACH text is greater than or equal to the Local Threshold.

Thresholds Example

GlobalThreshold=5 and LocalThreshold=2

```
text1: bi-gram X occurs 2 times
text2: bi-gram X occurs 3 times
text3: bi-gram X occurs 1 time
```

The bigram "x" qualifies for GlobalThreshold: 2+3+1 >= 5(GlobalThreshold), but it doesn't qualify for LocalThreshold, because for text3 it occurs 1<2 (LocalThreshold) times.

Bi-gram list

Here is an example.

fireman.txt

```
John is a fireman.
```

Bi-Grams:

```
John,is
is,a
a,fireman
```

Bi-grams List using Delete List and Generalization Thesaurus

This is an example of how a Delete List and Generalization Thesaurus can affect the final bi-gram list.

associations.txt

```
John Doe is actively involved in several industry
and civic associations.
```

associationsDeleteList.txt

```
is, in, and
```

associationsThesaurus.csv

```
John Doe,John_Doe
industry,business
civic associations,business
```

Using just the Delete List:

```
John Doe actively involved several industry civic
associations
```

The bi-grams list:

```
John,Doe
Doe,actively
actively,involved
involved,several
several,industry
industry,civic
civic,associations
```

Using just the Generalization Thesaurus:

```
John_Doe is actively involved in several business
and business
```

The bi-grams list:

```
John_Doe,is
is,actively
actively,involved
involved,in
in,several
several,business
business,and
and,business
```

Using both the Delete List and the Generalization Thesaurus:

```
John_Doe actively involved several business business
```

The bi-grams list:

```
john_Doe,actively
actively,involved
involved,several
several,business
business,business
```

Bi-Gram Chart

The sample text and following chart show the relationship of frequency and relative frequency of the concepts in the text.

businessLeader.txt

```
John Doe is a business leader. John Doe is a
president of the John Doe business.
```

Delete the noise from the text.

businessLeaderDeleteList.txt

```
is a of the
```

Both **John** and **Doe** have a frequency of **3**. The bigram **John Doe** also have a frequency of **3**. This shows these concepts are important as both individual words and the bigram they create.

Words	Frequency	Relative Frequency	Relative Percentage
John	3	1	.3
Doe	3	1	.3

Business	2	.67	.2
Leader	1	.33	.1
President	1	.33	.1
Total Words	10		

Bi-Grams	Frequency	Relative Frequency	Relative Percentage
John Doe	3	1	.37
Doe business	2	.67	.25
business leader	1	.33	.12
Doe president	1	.33	.12
president John	1	.33	.12
Total bi-grams	8		8

5 MAR 10

Concept Lists

Description

A Concept List is all the concepts of one individual file.

Using a Concept List a text can be refined using other functions such as a Delete List (to remove unnecessary concepts) and Generalization Thesaurus (to combine n-grams into single concepts).

Example:

johnDoeInc.txt

```
John Doe works at John Doe Inc.
```

Concept List:

```
John, Doe, works, at, John, Doe, Inc
```

johnDoeIncDeleteList.txt

```
at
```

Concept List after Delete List applied. The concept **at** is now missing.

```
John, Doe, works, John, Doe, Inc
```

johnDoeIncGenThes.csv

```
John Doe Inc,John_doe_inc
John Doe,john_doe
```

After applying Generalization Thesaurus the concept list has fewer concepts but they are more meaningful. **John** and **Doe** are combined into the person's name **John_Doe** as are the three individual concepts **John, Doe, & Inc.** into the name of the **John_Doe_Inc.**.

```
john_doe
works
john_doe_inc
```

NOTE : The order of the concepts in the Generalization Thesaurus is important. See Order of thesauri entries under Thesauri, Generalization for more information.

Information obtained from a Concept List

frequency
: The number of times a concept was found in a file

relative_frequency
: The frequency of any concept divided by the highest value obtained of any frequency.

gram_type

tf-idf
: term frequency–inverse document frequency - a statistical measure used to evaluate how important a word is to a document

Description

The Feature Selection creates a list of concepts as a TF*IDF (Term Frequency by Inverse Document Frequency) in descending order. This list can be used to determine the most important concepts in a file.

Date Styles

AutoMap understands certain styles of dates as shown below.

With the **month day, year** AutoMap detects the full date unless the day contains the numerical suffix.

```
January 1, 2009 => January 1, 2009, date
January 2nd, 2009 => January 2, date (the year was dropped)
```

The older military style date (with the abbreviated month) of **day month year** were all detected as currency. The modern **day month year** (with fully spelled out month) is detected as a date but drops the day.

```
1 FEB 09 => 1 FEB, currency
2 FEB 2009 => 2 FEB, currency
03 FEB 09 => 03 FEB, currency
04 FEB 2009 => 04 FEB, currency
5 February 2009 => February 2009,date dropped the day
```

The completely numerical style of date is detected as a number.

```
090301 => no entry
20090302 => no entry
```

the first one went undetected but the last three were correctly spotted as dates.

```
2009/4/1 => no entry
2009/04/2 => 2009/04, date (the day was dropped)
2009/4/03 => 2009/4/03, date
2009/04/04 => 2009/04/04, date
```

All detected as dates though some dropped off the year.

```
1/5/2009 => 1/5/2009, date
02/5/2009 => 02/5/2009, date
```

3/05/2009 => 3/05, date (the year was dropped)
04/05/2009 => 04/05/2009,date

All three detected as dates though some dropped the year.

June 1d, 2009 => June 1, date (the year was dropped)
June 2nd, 2009 => June 2, date (the year was dropped)

Both detected as dates but both dropped the day.

1 July 2009 => July 2009, date (the day was dropped)
02 July 2009 => July 2009, date (the day was dropped)

17 MAY 10

Delete Lists

Description

A Delete List is a list of concepts to be removed from a repository of text files. It is primarily used to reduce the number unnecessary concepts. By reducing the number of concepts being processed, run times are decreased and semantic networks (Kaufer & Carley. 1993) are easier to understand. This also helps in the creation of a semantic network in reducing the number of superficial nodes in ORA.

You can create Delete Lists for each set of files. This allows you to better refine the final output.

There are two types of adjacency: direct and rhetorical. The use of either one will be dictated by your need to maintain the original distance between concepts.

Points to Remember

The Delete List is **NOT** case sensitive. **He** and **he** are considered the same concept. Placing either one in the Delete List will move all instances.

You can create Delete Lists from a text editor or use the tools in AutoMap to assist in creating a specially-tailored Delete List.

All Delete Lists can be edited.

Multiple Delete Lists can be used on the same set of files.

Any Delete List can be saved and used for any other text files.

Adjacency

Direct Adjacency

This removes the concepts from the list totally. The concepts on either side then become adjacent to each other. This **does** affect the spacing between concepts.

tedDeleteList.txt

```
in, the, of, he, on, a, it
```

ted.txt

```
Ted lives in the United States of America. He lives
on a dairy farm. He considers it a good life. Would
he ever consider leaving?
```

Direct Adjacency

```
Ted lives United States America. He lives dairy
farm. He considers good life. Would he ever consider
leaving?
```

In the original text is the sentence: **He lives on a dairy farm.** After the deletion the concepts on a are removed and the concepts **lives dairy** are now adjacent.

Rhetorical Adjacency

This removes the concepts but inserts a spacer **xxx** within the text to maintain the original distance between all concepts of the input file. This **does not** affect the spacing between concepts.

tedDeleteList.txt

```
in, the, of, he, on, a, it
```

ted.txt

```
Ted lives in the United States of America. He lives
on a dairy farm. He considers it a good life. Would
he ever consider leaving?
```

Rhetorical Adjacency

```
Ted lives xxx xxx United States xxx America. He
lives xxx xxx dairy farm. He considers xxx xxx good
life. Would he ever consider leaving?
```

NOTE : xxx means that the concept is temporarily deleted and so is not in the current analytical focus.

In this example the same two words, **on a**, are removed from the original text. But with rhetorical adjacency spacers are inserted into the text. These two spacers maintain the exact distance between concepts as the original text. The results shows that there are two concepts between **Lives** and **dairy** but the substitution removes the actual concept from the result.

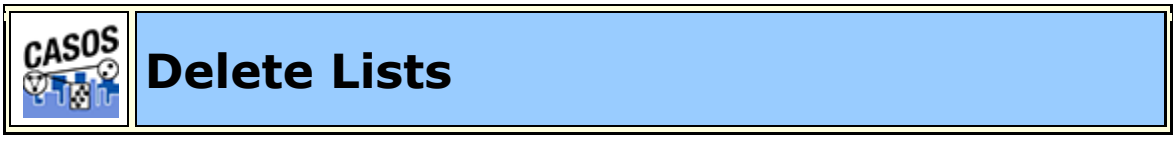
Reasons NOT to use a Delete List

For the most part using a Delete List on a file is a good idea. It removes many concepts that are unnecessary as they do not affect the meaning of the major concepts. But in some style of documents the meaning of two bi-grams could be drastically affected by two seemingly useless words. Most Delete Lists contain the concepts the and a. These two definite articles usually do not change the meaning of the text. But in some instances the meaning could be very substantial.

In a Field Operations manual there is a definite difference between the terms **a response** and **the response**. It is subtle, but very important.

Before using a Delete List, make sure that the words included do not change the meaning of the concepts surrounding them.

14 JAN 10



Excel when reading in a flat file (i.e. txt or .csv) is sensitive to the kind of delimiter used. In the American version of excel, it assumes that a comma or tab is used to separate columns. In other versions, it often assumes that a semicolon or tab is used to separate columns. This is because in many other languages the comma is used as a period in showing the price of items.

AutoMap and ORA export data as comma separated and can import comma separated. This means if you are reading into or reading from a non-American version of excel you may have problems.

Reading in the.csv file into Excel that uses something other than commas will cause the data to appear as a set of text in column A. There are two ways to fix this.

First read the file into a text editor and globally change all the delimiting characters to commas.

Second read the file into excel and use the **Text to Columns** function and use a different delimiter.

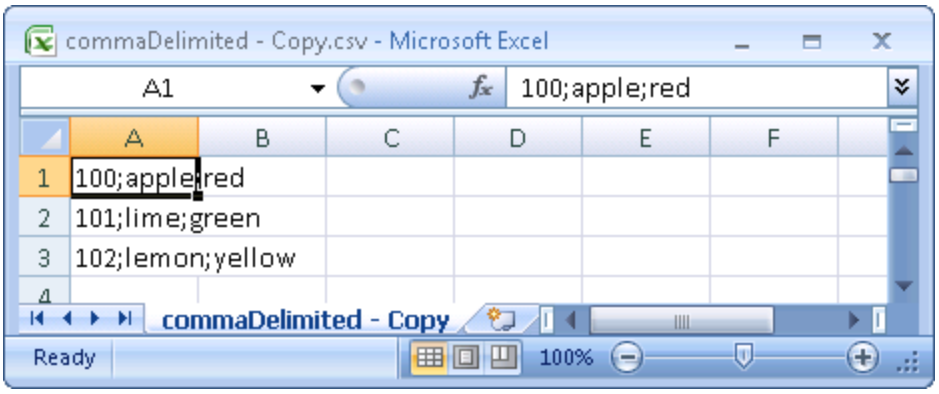
Excel Example

Let's start with a simple three line file using semicolons as delimiters.

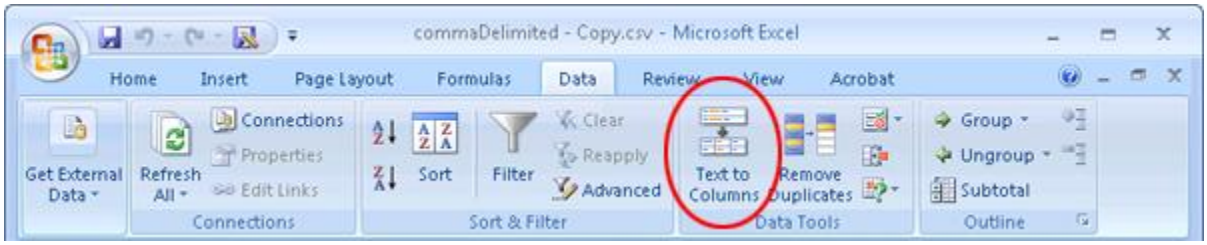
NOTE : Remember, data can be separated with a variety of characters. This procedure allows you to import data with any of them.

```
100;apple;red
101;lime;green
102;lemon;yellow
```

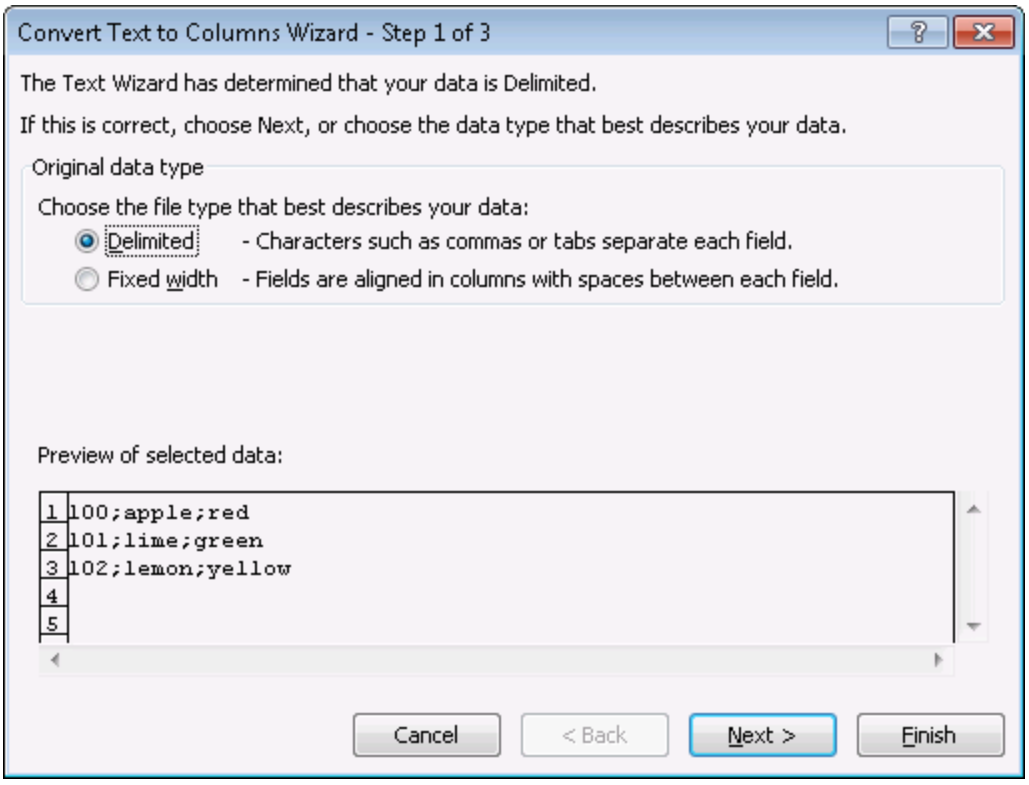
Opening this file in Excel will place each line of text into a single cell. You need to separate this into individual columns.



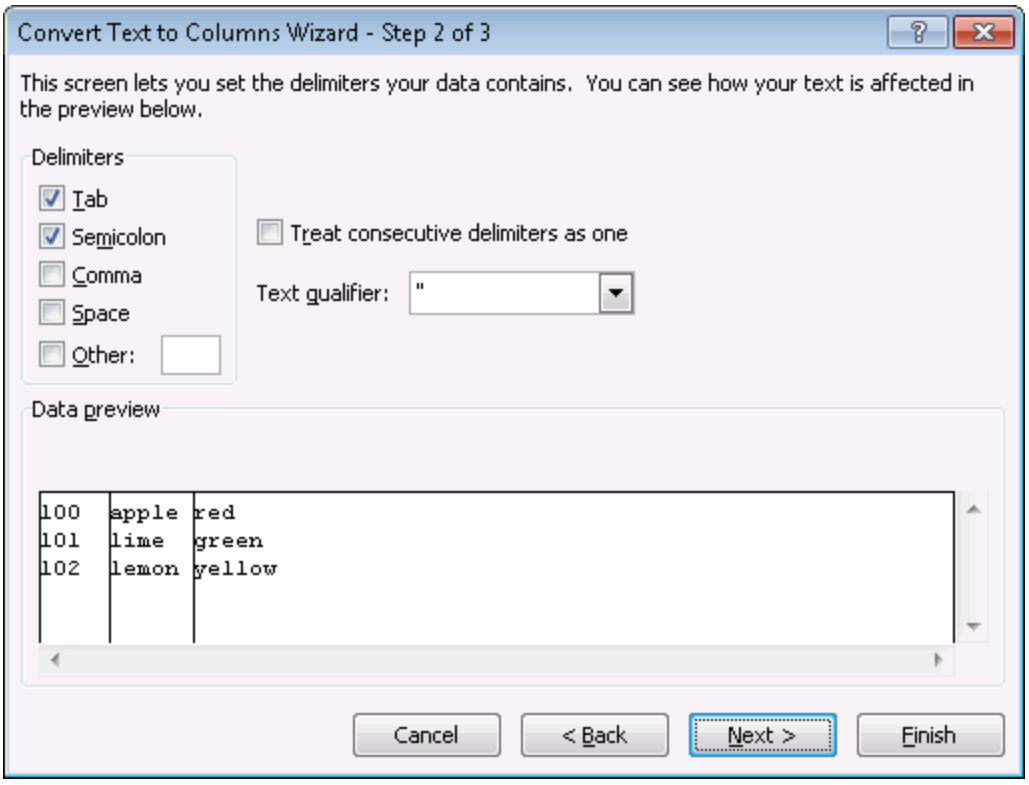
Select the cells you want to convert then click the Data tab. Click the **Text to Columns** function.



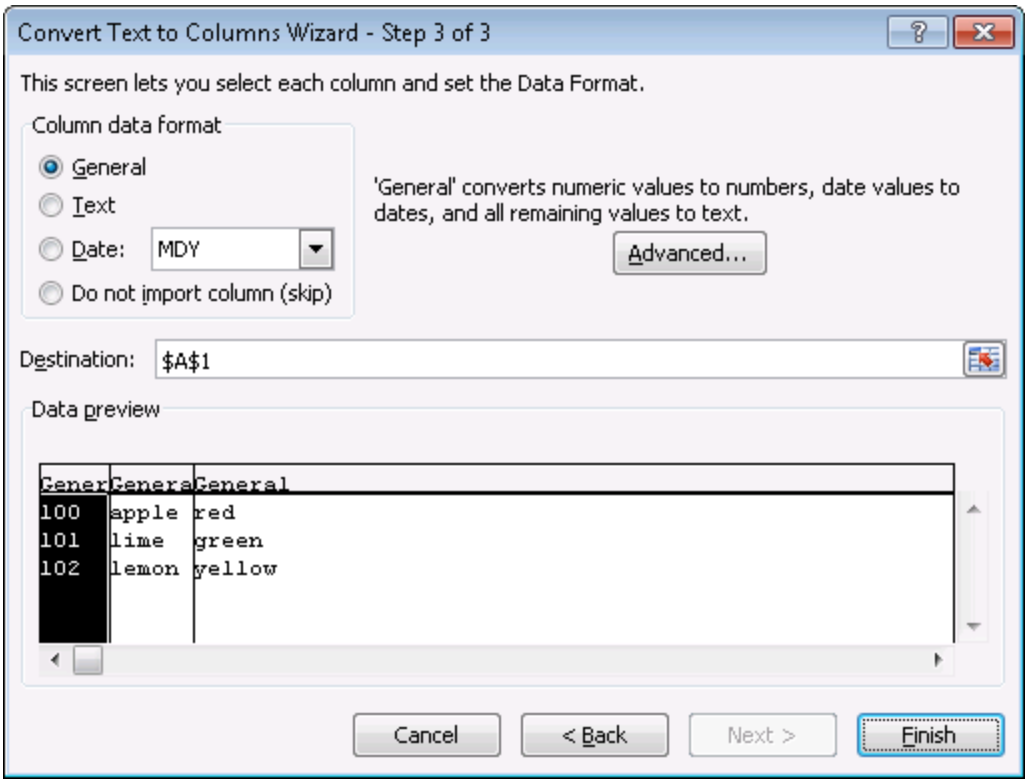
This brings up the **Convert Text to Columns Wizard**. Make sure the **Delimited** radio button is selected. Then click [**Next >**].



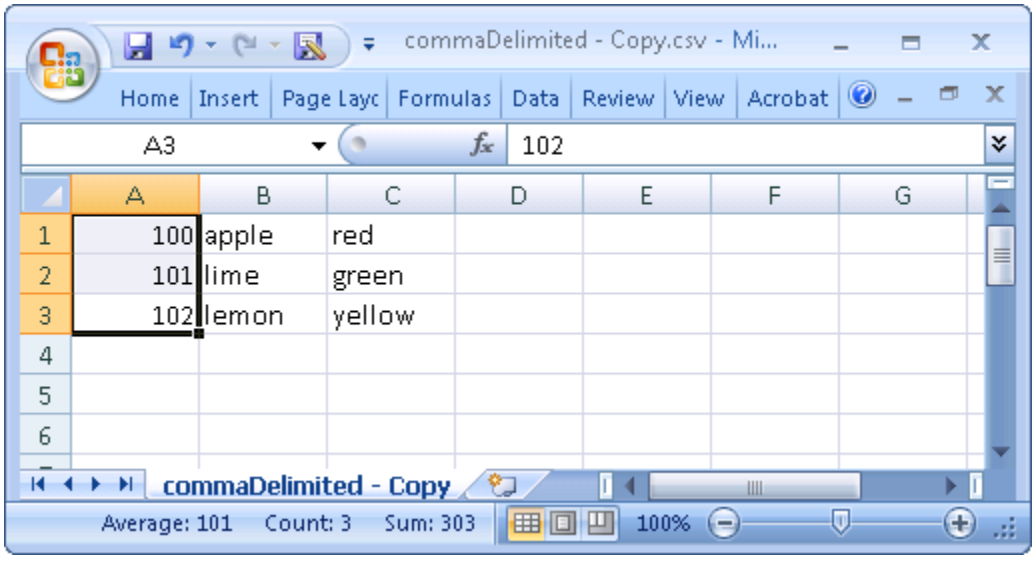
Under the **Delimiters** header make sure the **Semicolon** box contains a check mark. In the **Data preview**area it will show you what your data will look like after the conversion. Click [**Next >**].



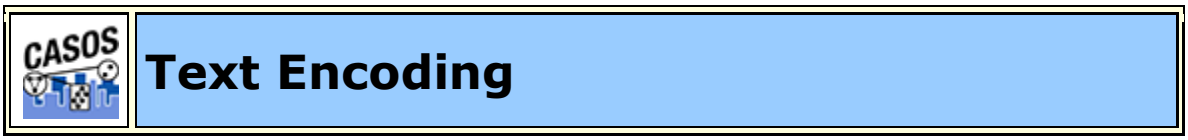
Lastly you can do some final tweaks to how your data will be converted if you want such as your data format or a different destination. In this example we want the data to be inserted back into the original starting cell. Click [**Finish >**].



Each piece of data now resides in a separate cell.



The new file can now be imported into ORA through the Data Import Wizard.



Text Encoding

Description

A character encoding system consists of a code that pairs a sequence of characters from a given character set (sometimes incorrectly referred to as code page) with something else, such as a sequence of natural numbers, octets or electrical pulses, in order to facilitate the transmission of data (generally numbers and/or text) through telecommunication networks and/or storage of text in computers.

UTF-8 : It is able to represent any character in the Unicode standard, yet is backwards compatible with ASCII. UTF-8 encodes each character (code point) in 1 to 4 octets (8-bit bytes), with the single octet encoding used only for the 128 US-ASCII characters. See the Description section below for details.

NOTE : If empty boxes appear in the text this is an indication the text is using the Microsoft version of UTF-8 instead of the standard encoding.

Western : A standard character encoding of the Latin alphabet. It is less formally referred to as Latin-1. It was originally developed by the ISO, but later jointly maintained by the ISO and the IEC. The standard, when supplemented with additional character assignments (in the C0 and C1 ranges: 0x00 to 0x1F and 0x7F, and 0x80 to 0x9F), is the basis of two widely-used character maps known as ISO-8859-1 (note the extra hyphen) and Windows-1252.

UTF-16 : (Unicode Transformation Format) is a variable-length character encoding for Unicode, capable of encoding the entire Unicode repertoire. The encoding form maps each character to a sequence of 16-bit words. Characters are known as code points and the 16-bit words are known as code units. For characters in the Basic Multilingual Plane (BMP) the resulting encoding is a single 16-bit word. For characters in the other planes, the encoding will result in a pair of 16-bit words, together called a surrogate pair. All possible code points from U+0000 through U+10FFFF, except for the surrogate code points U+D800-U+DFFF (which are not characters), are uniquely mapped by

UTF-16 regardless of the code point's current or future character assignment or use.

GB2312 : The registered internet name for a key official character set of the People's Republic of China, used for simplified Chinese characters. GB abbreviates Guojia Biaozhun, which means national standard in Chinese.

Big5 : The original Big5 character set is sorted first by usage frequency, second by stroke count, lastly by Kangxi radical. The original Big5 character set lacked many commonly used characters. To solve this problem, each vendor developed its own extension. The ETen extension became part of the current Big5 standard through popularity.

NOTE : AutoMap uses the **Hard Return** to designate paragraph breaks.

Text Direction

Languages can be written either left-to-right (LTR) or right-to-left (RTL). The majority of languages use a LTR syntax. The most notable RTL languages are Arabic and Hebrew.

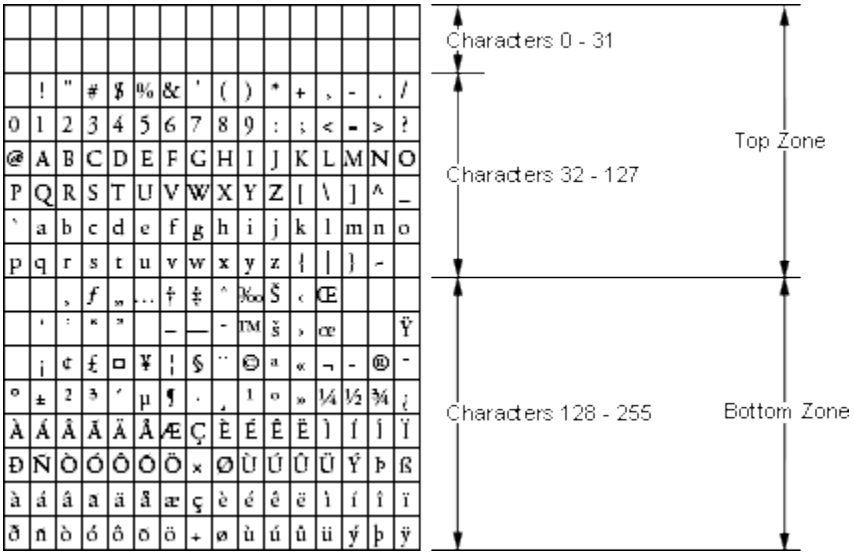
Single-Byte Fonts

Each character in a font uses a keystroke or combination of keystrokes to produce the character. Fonts based on most Western languages will have 256 possible characters. Each character in these fonts is one-byte in length. A one-byte character can have one of 256 possible values.

In a font, each character is arranged in a specific order. This is called a font's **encoding**, which is explained in more detail below. The computer uses this information to tell which character to display or print when a key is pressed. The spaces occupied by characters are called **slots**. Each slot represents a different keypress. For example, if you were working on a word-processing document and you were to hold the Shift key while pressing the letter "A" on the keyboard, you would see the letter **A** appear on the screen.

If you will notice, in the character map below, there are 2 **zones** - the top zone and the bottom zone. The top zone has characters

0 through 127 and the bottom zone has characters 128 through 255. The point to this is that characters occupying slots 32 through 127 in the top zone are identical in both Macintosh and Windows operating systems. Characters 0 through 31 (the "lower 32") are usually reserved for the operating system. The slots in the bottom zone (the **extended characters**) are different. These are the ones that will usually cause the problems.



NOTE : A font's encoding is simply a lookup table (an index) which is used to translate computer codes into the characters in the font.

13 MAY 09

Text Encoding Table

These tables include the text encodings that AutoMap is capable of importing.

Basic Encoding Set (contained in lib/rt.jar)

ISO-8859-1	ISO-8859-2	ISO-8859-4	ISO-8859-5
ISO-8859-7	**ISO-8859-9**	**ISO-8859-13**	**ISO-8859-15**
KOI8-R	US-ASCII	UTF-8	UTF-16

UTF-16BE	**UTF-16LE**	**windows-1250**	**windows-1251**
windows-1252	windows-1253	windows-1254	windows-1257

Extended Encoding Set (contained in lib/charsets.jar)

Big5	Big5-HKSCS	EUC-JP	EUC-KR
GB18030	**GB2312**	**GBK**	**IBM-Thai**
IBM00858	IBM01140	IBM01141	IBM01142
IBM01143	**IBM01144**	**IBM01145**	**IBM01146**
IBM01147	IBM01148	IBM01149	IBM037
IBM1026	**IBM1047**	**IBM273**	**IBM277**
IBM278	IBM280	IBM284	IBM285
IBM297	**IBM420**	**IBM424**	**IBM437**
IBM500	IBM775	IBM850	IBM852
IBM855	**IBM857**	**IBM860**	**IBM861**
IBM862	IBM863	IBM864	IBM865
IBM866	**IBM868**	**IBM869**	**IBM870**
IBM871	IBM918	ISO-2022-CN	ISO-2022-JP
ISO-2022-KR	**ISO-8859-3**	**ISO-8859-6**	**ISO-8859-8**
Shift_JIS	TIS-620	windows-1255	windows-1256
windows-1258	**windows-31j**	**x-Big5_Solaris**	**x-euc-jp-linux**
x-EUC-TW	x-eucJP-Open	x-IBM1006	x-IBM1025
x-IBM1046	**x-IBM1097**	**x-IBM1098**	**x-IBM1112**
x-IBM1122	x-IBM1123	x-IBM1124	x-IBM1381
x-IBM1383	**x-IBM33722**	**x-IBM737**	**x-IBM856**
x-IBM874	x-IBM875	x-IBM921	x-IBM922
x-IBM930	**x-IBM933**	**x-IBM935**	**x-IBM937**

x-IBM939	x-IBM942	x-IBM942C	x-IBM943
x-IBM943C	**x-IBM948**	**x-IBM949**	**x-IBM949C**
x-IBM950	x-IBM964	x-IBM970	x-ISCII91
x-ISO2022-CN-CNS	**x-ISO2022-CN-GB**	**x-iso-8859-11**	**x-JISAutoDetect**
x-Johab	x-MacArabic	x-MacCentralEurope	x-MacCroatian
x-MacCyrillic	**x-MacDingbat**	**x-MacGreek**	**x-MacHebrew**
x-MacIceland	x-MacRoman	x-MacRomania	x-MacSymbol
x-MacThai	**x-MacTurkish**	**x-MacUkraine**	**x-MS950-HKSCS**
x-mswin-936	x-PCK	x-windows-874	x-windows-949
x-windows-950			

07 OCT 09

File Formats

Description

There are many types of text formats available. Only the text format with the .txt extension works correctly in AutoMap. If your data is in any other format it must be converted before using it in AutoMap.

Thesauri Format

Thesauri Format :
conceptFrom,conceptTo

Master Thesauri Format

Master Thesauri Format :
conceptFrom,conceptTo,metaOntology,MetaName

The Master Thesauri combines the generalization **(from and to)**, the metanetwork **(to and meta)**.

Marking for the Delete List is represented when the character [**#**] is placed in the meta column.

Any item which contains  as part of it's icon is related to a Master Thesauri funtion.

metaOntology is one of the ora types: **agent, organization, location, event, knowledge, resource, task.**

*See **Content Overview => Ontology** for more information.*

Example

conceptFrom : United State of America

conceptTo : USA

metaOntology : location

04 JAN 11

Format Case

Description

Format Case changes the output text to either all lower or upper case.

Example

Sentence case

Only the first word of the sentence and proper nouns are capitalized.

My name is John Smith and I live in the USA.

Lower case

All letters are lowercase, even proper nouns.

my name is john smith and i live in the usa.

Upper case

All letters are uppercase, even proper nouns.

MY NAME IS JOHN SMITH AND I LIVE IN THE USA.

Title case

The first letter of every word is capitalized.

My Name Is John Smith And I Live In The USA.

NOTE : The problem with converting text is it disables the ability of Parts of Speech to correctly identify certain parts - such as Proper Nouns.

13 MAY 09

Master Format

The Master Format was introduced to AutoMap to give files more versatility than the legacy format. This format is used throughout all support files in AutoMap including Generalization Thesauri, Meta-Network Thesauri, and Delete Lists.

The Master Format file contains the following information:

- **Concept From :** The term contained in your text which AutoMap will search for.
- **Concept To :** The term that will replace the Concept From when found.
- **MetaOntology :** The category (if any) to use for the term found. This will be agent, knowledge, resource, task, event, organization, location, role, action, attribute, when. Information about ontology can be found on **the Ontology Page**
- **MetaName :** For future use.

23 MAR 11

Meta-Network Thesaurus

Description

The Meta-Network (Carley, 2002) Thesaurus maps key words in a text file with the categories to create a Meta-Network. This can be done at any step of the process but it is suggested that a Delete List and/or General Thesaurus is run previously. This makes sure that unnecessary terms aren't mapped into the network.

It is primarily used for preparing a file for importing into ORA and the creation of a semantic network to analyze. ORA looks for Nodes and NodeSets. This process groups those concepts into the NodeSets used by ORA.

A Meta-Network Thesaurus associates concepts with the following meta-network categories: Agent, Knowledge, Resource, Task/Event, Organization, Location, Action, Role, Attribute, Any user-defined category (as many as the user defines).

Named Entities

Description

Named-Entity Recognition allows you to retrieve proper names numerals, and abbreviations from texts.

Items it Detects:

- Single words that are capitalized (e.g. Copenhagen).
- Adjacent words that are capitalized (e.g. The New York City Police Department).
- A string of adjacent words that are capitalized, but can be intervened by one non-capitalized word. The first and the last word in this string are capitalized (e.g. Canadian Department of National Defense).

13 MAY 09

Networks

Description

AutoMap is concerned with a variety of different types of networks. Below is a chart showing the various types of networks and how they interact with each other.

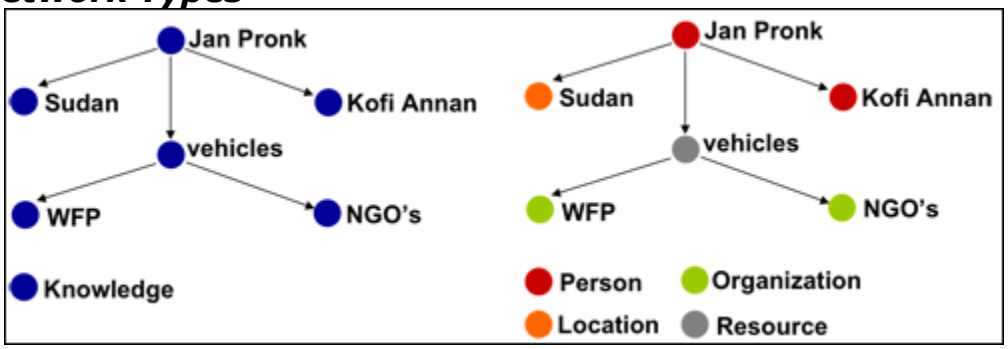
Items it Detects:

Agent	**Interaction Network** **Who knows who Structure**	**Knowledge Network** **Who knows what-Culture**	**Assignment Network** **Who is assigned to what-Jobs**	**Employment Network** **Who works where-Demography**
Knowledge		**Information Network** **What informs what-Data**	**Requirements Network** **What is needed to do what-Needs**	**Competency Network** **What knowledge is where-Culture**
Tasks			**Precedence Network** **What needs to be done before what-Operations**	**Industrial Network** **What tasks are done where-Niche**
Organizations				**Inter-organizational Network** **Which organizations work with which-Alliances**

Meta-Network	Agent	Knowledge	Resource	Task/Event	Organization	Location
agent	**Social nw**	**Knowledge nw**	**capabilities nw**	**assignment nw**	**membership nw**	**agent location nw**
Knowledge		**Information nw**	**Training nw**	**Knowledge requirement nw**	**Organizational knowledge nw**	**Knowledge location nw**
Resource			**Resource**	**Resource**	**Org.**	**Resource**

			nw	requirement nw	Capabilities nw	location nw
Task/Event				Precedence nw	Org. Assignment nw	Task/Event nw
Organization					Interorg. nw	Org. locaion nw
Location						Proximity nw

Network Types



One Mode Network

Represent reality that people have in their minds and use to make sense of their surroundings.

Semantic Networks as Mental Models: Single Mode Networks are usually Semantic Networks. Nodes are not distinguished in any way. In the example all nodes are classed as knowledge. Represent reality that people have in their minds and use to make sense of their surroundings. Cognitive constructs that reflect the subjects' knowledge and information about a certain topic.

Multi-Mode Network

Identification and classification of all relevant instances of node and edge classes from texts as efficiently and accurately as possible.

Which agent or group is located where, has access to what resources, possesses what knowledge, is involved in what tasks, has what personal characteristics, ... ?

Nodes are classified by category and ORA can use these classifications for analysis. On the right nodes are classed as person, location, organization or resource.

Multi-Mode networks are **Ontologically coded** socio-technical networks which classify relevant nodes according to some ontology or taxonomy.

26 JUN 09

Ontology

Description

AutoMap gives the user the ability to **define their own ontology**.

Instead of just refering to the people involved as **agents** you could differiante them as **good_guys** and **bad_guys**.

Using a new ontology with ORA

NOTE : Although you can define any node to be defined by specialty tags ORA will not understand these new definitions in it's reporting functions. When producing a report, such as Emmergent Leader, ORA will look at nodes tagged by **agent** only.

Standard MetaNetwork categories

Below are the standard tags used in ORA for it's reporting.

- **action :** driving to the mall, eating lunch. Used as a verb.
- **agent :** A person, group, organization, or artificial actor that has information processing capabilities. All whos are agents whether they be a person in a group, a group within an organization, or the organization itself (e.g. President Barack Obama, the shadowy figure seen outside the building, or the Census bureau). It is up to the user's discretion what sub-category to place these agents in.

- **attribute :** Information about the specifics of the agents. These are usually traits that agents have in common, each can be slightly different (e.g. visible traits like hair colour or intangible traits like religious beliefs).
- **event :** Something that happens, especially something of importance. Events are usually thought of as a public occasions but they can also be clandestine meetings. The number of agents can range in the thousands or as few as two agents (e.g. Christmas in Times Square or dinner with friends).
- **knowledge :** Information learned such as a school lecture or knowledge learned from experience (e.g. Excellent knowledge of the periodic table or "I know what you did last summer").
- **location :** An actual physical place. This could be a room in a building, a city, or a country (e.g. Pittsburgh, PA or my living room).
- **organization :** A group of agents working together for a common cause (e.g. The Red Cross or the local chess club).
- **resource :** Can be either a physical or intangible object. Anything that can be used for the completion of a job. (e.g. Use a car to drive from point A to point B or use money from a bank account to fund something).
- **role :** An agents role can be defined as their job for their employer or the part they serve during an event.
- **task :** A task is part of a set of actions which accomplish a job, problem or assignment. Task is a synonym for activity although the latter carries a connotation of being possibly longer duration (e.g.)
- **when :** Referring to time or circumstances. Can be as broad as a year or as pinpoint as the exact time of a particular day (e.g. Last year or 2:33 PM on March 1st, 2009).

Specific vs. Generic

The difference between Specific and Generic can be simply put as the difference between **a** and **the**.

Parts of Speech

Description

Parts of Speech assigns a single best **Part of Speech**, such as noun, verb, or preposition, to every word in a text.

While many words can be unambiguously associated with one tag, (e.g. computer with noun), other words match multiple tags, depending on the context that they appear in.

Example : Wind, for example, can be a noun in the context of weather, and can be a verb that refers to coiling something.) DeRose (DeRose, 1988) reports that over 40% of the words are syntactically ambiguous.

Parts of Speech is often necessary before other functions are performed specifically when creating a Meta-Network (Carley, 2002). This Parts of Speech tagger is based on the **Hidden Markov Model**.

The Hidden Markov Model

A Hidden Markov model (HMM) is a statistical model in which the system being modeled is assumed to be a Markov process with unknown parameters; the challenge is to determine the hidden parameters from the observable data. The extracted model parameters can then be used to perform further analysis, for example for pattern recognition applications. An HMM can be considered as the simplest dynamic Bayesian network.

Penn Tree Bank (PTB) Parts of Speech Table

CC	Coordinating conjunction	**PRP$**	Possessive pronoun
CD	Cardinal number	**RB**	Adverb
DT	Determiner	**RBR**	Adverb, comparative
EX	Existential there	**RBS**	Adverb, superlative
FW	Foreign word	**RP**	Particle

IN	Preposition or subordinating conjunction	**SYM**	Symbol
JJ	Adjective	**TO**	to
JJR	Adjective, comparative	**UH**	Interjection
JJS	Adjective, superlative	**VB**	Verb, base form
LS	List item marker	**VBD**	Verb, past tense
MD	Modal	**VBG**	Verb, gerund or present participle
NN	Noun, singular or mass	**VBN**	Verb, past participle
NNS	Noun, plural	**VBP**	Verb, non-3rd person singular present
NNP	Proper noun, singular	**VBZ**	Verb, 3rd person singular present
NNPS	Proper noun, plural	**WDT**	Wh-determiner
PDT	Predeterminer	**WP**	Wh-pronoun
POS	Possessive ending	**WP$**	Possessive wh-pronoun
PRP	Personal pronoun	**WRB**	Wh-adverb

Aggregate Parts of Speech

The PTB divides verbs into six subgroups (base form verbs, present participle or gerund verbs, present tense not 3rd person singular verbs, present tense 3rd person singular verbs, past participle verbs, past tense verbs). In some applications you might want to aggregate these into one verb group. Also, for certain purposes, the union of all prepositions, conjunctions, determiners, possessive pronouns, particles, adverbs, and interjections could be collected into one group that represents irrelevant terms.

Aggregation of PTB Categories

Aggregated Tag	Meaning	Number of Categories in PTB	Instances in PTB

IRR	Irrelevant term	16	409,103
NOUN	Noun	2	217,309
VERB	Verb	6	166,259
ADJ	Adjective	3	81,243
AGENTLOC	Agent	1	62,020
ANA	Anaphora	1	47,303
SYM	Noise	8	36,232
NUM	Number	1	15,178
MODAL	Modal verb	1	14,115
POS	Genitive marker	1	5,247
ORG	Organization	1	1,958
FW	Foreign Word	1	803

Noise

Typically, text data includes various types of noise in varying quantity. What precisely qualifies as noise and how much of it will be normalized or eliminated depends on the goal, resources, and researcher. A list can be created which dictates the parameters of what can be included as POS. All tokens that are or comprise any symbol not listed above can be considered as noise.

Why is determining what is noise important? People are typically not interested in predicting tags for symbols, but only for what is typically considered as content. Another point is processing noise takes time and resources. Removing noise first speeds up the process.

johnIsAFireman.txt

```
John is a Fireman in lower Manhattan in New York
City. John was there at the Twin Towers on that day
in September.
```

This text can be tagged in two distinct ways: PTB and Aggregated. These POS lists are also done before any other pre-processing such as a Generalization Thesaurus so New, York, and City aren't all tagged individually.

PTB Tagging

```
John/NNP is/VBZ a/DT Fireman/NN in/IN lower/JJR
manhattan/NN in/IN New/NNP York/NNP City/NNP ./.
John/NNP was/VBD there/RB at/IN the/DT Twin/JJ
Towers/NN on/IN that/DT day/NN in/IN September/NNP
./.
```

The aggregated tagging combines many PTB tags into one. In PTB is/VBZ and was/VBD are combined and both are tagged as /VERB.

Aggregated Tagging

```
John/AGENTLOC is/VERB a/IRR Fireman/NOUN in/IRR
lower/ADJ manhattan/NOUN in/IRR New/AGENTLOC
York/AGENTLOC City/AGENTLOC ./. John/AGENTLOC
was/VERB there/IRR at/IRR the/IRR Twin/ADJ
Towers/NOUN on/IRR that/IRR day/NOUN in/IRR
September/AGENTLOC ./.
```

23 SEP 09

Relation Types

Description

This module identifies the type of relation between two entities and extracts networks based on those types. The relations are extracted from the sentences that consist of two noun phrases connected by a verb phrase. The relation type is described by the main verb, and the noun the arguments of the relation are given by the noun phrases.

For example, the module identifies the following sentence as a candidate for extracting the relation it describes: "Kofi Annan visited Damascus". The entities in the relation are "Kofi Annan" and "Damascus", while the type of the relation is "to visit".

The module creates sub-networks from the extracted relations, grouping them according to the semantically similarity of their types. For instance, relations including the following verbs are grouped in the same subnetwork: travel, visit and tour; as well as their conjugated versions, such as visits, visiting or toured. The verbs that are not included into one of the groups, are added to a general category.

This module needs to be executed on the texts without any pre-processing steps. Internally, it depends on a Part-of-Speech tagger (POS), a shallow parser (chunker). and a named entity recognizer. For the first two, the implementations included in OpenNLP are used, while the named entities are identified using the CRF entity tagger from AutoMap.

List of networks

The following is the list of network types extracted with this module:

```
id - Name
accuse - Accuse
arrested - Arrested
attack - Attack
call - Call
collaborate - Collaborate
establish - Establish
export - Export
manager of - Manager of
meet - Meet
member of - Member of
plan - Plan
receive - Receive
support - Support
talk - Talk
visit - Visit
```

Execution

Inputs

The module requires the following inputs:

A list of input files, i.e. the current documents imported in AutoMap.

A meta-network thesaurus

Output files

The module generates two directories with the following structure:

Relations:

A set of CSV files describing the networks as follows:

concepts/reltypes.csv: A concepts lists file that includes the name and type of the identified concepts.

relations/reltypes.csv: A semantic lists file; each line represents a relation between two concepts and the type of this relation.

properties/reltypes.csv: An empty file. This file is created for keeping the compatibility with other modules.

Networks:

A directory containing a DynetML file that includes all the extracted subnetworks.

Script tags

The module adds the following two tags for AM3Script

<RelTypeTagger />

Processes the input documents with the results of the Part-of-Speech tagger (POS), the shallow parser (chunker) and the named entity recognizer. This tags invokes a module that takes as inputs the text files imported in Automap and creates files with the output of the taggers.

Attributes:

inputDirectory: Directory containing the text input files.

outputDirectory: Directory in which the output files will be created.

posModel: Path to the POS model file for OpenNLP. The recommended value is "$AUTOMAP_HOME/etc/opennlp/en-pos-maxent.bin"

chunkingModel: Path to the chunking module file for OpenNLP. The recommended value is "$AUTOMAP_HOME/etc/opennlp/en-chunker.bin"

crfDir: Path to the directory containing the configuration files of the CRF tagger.

<RelTypeExtract />

Extracts the relations from a processed set of files and generates the network definition files. This tag invokes a module that takes as inputs the output of <RelTypeTagger /> and creates the network definition files in CSV and DynetML formats.

Attributes:

inputDirectory: Directory containing the files created by <RelTypeTagger />.

outputDirectory: Directory in which the output files will be created.

thesaurus: Thesaurus file containing the concepts used in the extracted networks.

networks: List of the networks that will be extracted. It can be a list of names separated by commas (such as "accuse,talk"visit") or the operator "all" to extract all the networks.

clusters: File that defines how the relations are organized into groups. The recommended value is "$AUTOMAP_HOME/etc/reltypes/cluser-verbs.csv". This is a CSV file that contains 2 columns: word and clusterid. The first one is a verb and the second is the name of the assigned sub-network (see the list of networks). The following is an example of the contents of the file:

```
word,clusterid
criticise,talk
criticised,talk
```

```
criticizing,talk
remarks,talk
ask,talk
underlined,talk
```

Output directories:

This tag creates two output directories with the files describing the networks:

relations:

A set of CSV files describing the networks as follows:

concepts/reltypes.csv: A concepts lists file that includes the name and type of the identified concepts.

relations/reltypes.csv: A semantic lists file; each line represents a relation between two concepts and the type of this relation.

properties/reltypes.csv: An empty file. This file is created for keeping the compatibility with other modules.

networks:

A directory containing a DynetML file that includes all the extracted subnetworks.

Configuration options:

The configuration files are stored in the directory "$AUTOMAP_HOME/etc/reltypes".

cluster-defs.csv:

A CSV file with a list of network names and descriptions. This file is used to display the options of possible networks in the graphical user interface. It has two columns: clusterid, name; the first one corresponds to the internal name of the network, and the second one is human readable description. The following is an example of the contents of the file:

```
clusterid,name
accuse,Accuse
arrested,Arrested
```

```
support,Support
talk,Talk
visit,Visit
```

cluster-verbs.csv

File that defines how the relations are organized into groups. The recommended value is "$AUTOMAP_HOME/etc/reltypes/cluser-verbs.csv". This is a CSV file that contans 2 columns: word and clusterid. The first one is a verb and the second is the name of the assigned sub-network (see the list of networks). The following is an example of the contents of the file:

```
word,clusterid
criticise,talk
criticised,talk
criticizing,talk
remarks,talk
ask,talk
underlined,talk
```

The networks listed in both files must be the same; i.e. the file cluster-verbs.csv must contain all the networks listed in cluser-defs.csv, otherwise they will not be extracted.

Semantic Lists

Description

Semantic Lists contain pairs of concepts and their frequency in the chosen text file(s).

Direction

Uni-directional : Will only look forward in the text file for a relationship. Any concept that came before will be ignored.

Bi-Directional : Will attempt to find a relationship in either direction of the concept. Both are constrained by windowSize and textUnit.

```
agent1 xxx agent2 xxx agent3.
```

Using **uni-directional and a window size of 3** agent2 would have a relationship to agent3 but not agent1. Relationships can only look forward in the text.

Using **bi-directional and a window size of 3** agent2 would have a relationship to both agent3 and agent1

NOTE : Using bidirectional can substantially increase the size of the Semantic List. A file with 17 concepts and using a window of 2 produced a unidirectional Semantic List of 13 entries whereas the bidirectional Semantic List consisted of 26 entries.

Window Size

The distant concepts can be and still have a relationship to one another. Only concepts in same window can form statements. The window is defined in **textUnit**.

Text Unit

The text unit can be comprised of one of the following:

Sentence : a sentence is a grammatical unit of one or more words.

Word : A word is a unit of language that represents a concept which can be expressively communicated with meaning

Clause : A clause consists of a subject and a verb. There are two types of clauses: independent and subordinate (dependent).

> An **independent clause** consists of a subject verb and also demonstrates a complete thought: for example, `"I am sad"`.
>
> A **subordinate clause** consists of a subject and a verb, but demonstrates an incomplete thought: for example, `"Because I had to move"`.

Paragraph : A paragraph is indicated by the start of a new line. It consists of a unifying main point, thought, or idea accompanied by supporting details.

All : The entire text

Semantic Networks

Description

Semantic networks are knowledge representation schemes involving nodes and links between nodes. It is a way of representing relationships between concepts. The nodes represent concepts and the links represent relations between nodes. The links are directed and labeled; thus, a semantic network is a directed graph.

Directional

Uni-directional : will only look forward in the text file for a relationship. Any concept that came before will be ignored.

Bi-Directional : will attempt to find a relationship in either direction of the concept. Both are constrained by windowSize and textUnit.

```
agent1 xxx xxx agent2 xxx xxx agent3.
```

Using **uni-directional** agent2 would have a relationship to agent3 but not agent1. Relationships can only look forward in the text.

Using **bi-directional** agent2 would have a relationship to both agent3 and agent1./p> Window Size

The distant concepts can be and still have a relationship to one another. Only concepts in same window can form statements. The window is defined in **textUnit**.

Text Unit

The text unit can be comprised of one of the following:

Sentence : a sentence is a grammatical unit of one or more words.

Word : A word is a unit of language that represents a concept which can be expressively communicated with meaning

Clause : A clause consists of a subject and a verb. There are two types of clauses: independent and subordinate (dependent). An independent clause consists of a subject verb and also demonstrates a complete thought: for example, "I am sad." A subordinate clause consists of a subject and a verb, but demonstrates an incomplete thought: for example, "Because I had to move."

Paragraph : A paragraph is indicated by the start of a new line. It consists of a unifying main point, thought, or idea accompanied by supporting details.

All : The entire text

Example

dairyFarm.txt

```
Ted runs a dairy farm. He milks the cows, runs the
office, and cleans the barn.
```

Semantic Network parameters:

```
windowSize="2" textUnit="S" directional="U"
resetNumber="1"
```

Concept List:

```
concept, frequency, relative_frequency, gram_type
He,1,0.5,single
Ted,1,0.5,single
a,1,0.5,single
and,1,0.5,single
barn,1,0.5,single
cleans,1,0.5,single
cows,1,0.5,single
dairy,1,0.5,single
farm,1,0.5,single
milks,1,0.5,single
office,1,0.5,single
runs,2,1.0,single
the,3,1.5,single
```

Word List:

```
Ted, runs, a, dairy, farm, He, milks, the, cows,
runs, the, office, and, cleans, the, barn
```

Property List:

```
Number of Characters,79
Number of Clauses,4
Number of Sentences,2
Number of Words,16
```

Semantic Network csv:

```
concept, concept, frequency
He,milks,1
Ted,runs,1
a,dairy,1
and,cleans,1
cleans,the,1
cows,runs,1
dairy,farm,1
farm,He,1
milks,the,1
office,and,1
runs,a,1
runs,the,1
the,barn,1
the,cows,1
the,office,1
```

23 SEP 09

Process Sequencing

Description

When processing data it's important to consider the order which preprocessing functions are done. In some circumstances the output will not be what you expect.

Delete List and Generalization Thesaurus

In the example sentence the concept **the** is both as a stand alone concept and also as part of a title. The first instance is noise and can be eliminated but the second instance is part of the movie title.

rings.txt

```
Dave likes the movie The Lord of the Rings
```

So you create a Delete List and a Generalization Thesaurus to remove the unwanted concepts but conserve the movie title.

ringsDeleteList.txt

```
the
of
```

ringsGenThes.csv

```
The Lord of the Rings,The_Lord_of_the_Rings
```

Run the Delete List then Thesaurus

If the Delete List is applied first with a rhetorical adjacency the following is obtained. You can see that the title can no longer be replaced by the Generalization Thesaurus.

```
Dave likes xxx movie xxx Lord xxx xxx Rings.
```

The replacement in the Generalization Thesaurus is impossible to apply as the **of** and the **the** in the title have been deleted.

Run the Thesaurus then Delete List

But if the Generalization Thesaurus is applied first the title is replaced before the Delete List removes the noise.

```
Dave likes the movie The_Lord_of_the_Rings.
```

Then the Delete List can remove the other **unwanted** concepts.

```
Dave likes xxx movie The_Lord_of_the_Rings.
```

22 JUL 09

Stemming

Description

Stemming is a process for removing the more common morphological and inflectional endings from words in English. It detects inflections and derivations of concepts in order to convert each concept into the related morpheme. This assists in counting similar concepts in the singular and plural forms (e.g. plane and planes would normally be considered two terms). After

stemming, "planes" becomes "plane" and the two concepts are counted together.

This can be broken down into two subclasses, **Inflectional and Derivational**.

- **Inflectional** morphology describes predictable changes a word undergoes as a result of syntax (the plural and possessive form for nouns, and the past tense and progressive form for verbs are the most common in English). These changes have no effect on a word's **part-of-speech** (a noun still remains a noun after pluralizations).
- **Derivational** morphology may or may not affect a word's meaning (e.g.; '-ise', '-ship'). Although English is a relatively weak morphological language, languages such as Hungarian and Hebrew have stronger morphology where thousands of variants may exist for a given word. In such a case the retrieval performance of an IR system would be severely be impacted by a failure to deal with such variations.

K-STEM

KSTEM or Krovetz stemmer (Krovetz, 1995, a dictionary-based stemmer) : The Krovetz Stemmer effectively and accurately removes inflectional suffixes in three steps, the conversion of a plural to its single form (e.g. '-ies', '-es', '-s'), the conversion of past to present tense (e.g. '-ed'), and the removal of '-ing'. The conversion process firstly removes the suffix, and then though a process of checking in a dictionary for any recoding (also being aware of exceptions to the normal recoding rules), returns the stem to a word. This Stemmer is frequently used in conjunction with other Stemmers, making use of the advantage of the accuracy of removal of suffixes by this Stemmer. For the Krovetz stemmer, several customization options are offered:

K-STEM Example

tedInUSA.txt

```
Ted lives in the United States of America. He lives
on a dairy farm. He considers it a good life. Would
he ever consider leaving?
```

Text after K-Stemming:

```
Ted live in the Unite State of America. He live on a
dairy farm. He consider it a good life. Would he
ever consider leave?
```

Porter Stemming

The **Porter stemmer** uses the Porter Stemming algorithm. Additionally, it converts irregular verbs into the verb's infinitive.

Porter Example

tedInUSA.txt

```
Ted lives in the United States of America. He lives
on a dairy farm. He considers it a good life. Would
he ever consider leaving?
```

Text after Porter Stemming:

```
Ted live in the Unite State of America. He live on a
dairi farm. He consid it a good life. Would he ever
consid leav?
```

Languages for Porter Stemming

Each language's stems work differently. Failing to use the correct language files when stemming risks obtaining incorrect results.

Differences in Stemming

There is a difference in the way the Porter and K-Stem functions stem words: **consider(s) and dairy**.

Porter removes both the **er** and the **ers** from the words consider and considers. **K-Stem** removes the **s** from considers and both words end up as consider.

Porter changes the **y** in dairy to an **i** whereas **K-Stem** leaves the word untouched.

Stem Capitalized Concepts

Decide whether or not to stem capitalized words. This will include all proper nouns.

NOTE : If capitalized words are not stemmed then remember that the first word of each sentence will likewise not be stemmed.

Porter, M.F. 1980. *An algorithm for suffix stripping.* I 14 (3): 130-137.

Krovetz, Robert 1995. *Word Sense Disambiguation for Large Text Databases.* Unpublished PhD Thesis, University of Massachusetts.

5 MAR 10

Text Formats

Description

There are many types of text formats available. Only the text format with the .txt extension works correctly in AutoMap. If your data is in any other format it must be converted before using it in AutoMap.

Text Formats The only format AutoMap can read. Uses the .txt file extension.

Other text formats

- **ASCII :** (American Standard Code for Information Interchange) is the lowest common denominator. There are actually two ASCII codes. The original 128 character, 7-bit code and the expanded 256 character, 8-bit code.
- **CSV :**(Comma Separated Value) A file type that stores tabular data. The format dates back to the early days of business computing. For this reason, CSV files are common on all computer platforms.
- **EBCDIC :**(Extended Binary Coded Decimal Interchange Code) is an 8-bit character encoding used on IBM mainframe operating systems such as z/OS, OS/390, VM

and VSE, as well as IBM minicomputer operating systems such as OS/400 and i5/OS.

- **HTML :**(Hypertext Markup Language) The predominant markup language used for web pages. It is a text format but uses a tagging system which would be interrupted as concepts by AutoMap.
- **ISO/IEC 8859 :** Standard for 8-bit character encodings for use by computers.
- **RTF :**(Rich Text Format) A proprietary document file format developed by DEC in 1987 for cross-platform document interchange. Most word processors are able to read and write RTF documents.
- **UTF-8 :**(Uniform Transformation Format) It is able to represent any character in the Unicode standard, yet the initial encoding of byte codes and character assignments for UTF-8 is backward compatible with ASCII. For these reasons, it is steadily becoming the preferred encoding for e-mail, web pages, and other places where characters are stored or streamed.
- **XML :**(Extensible Markup Language) A general purpose markup language that allows users to define their own tags.

04 JAN 11

Text Properties

Description

Outputs information regarding the currently loaded files. AutoMap writes one file for each file currently loaded.

milkAndCookies.txt

```
Dave wants milk and cookies. He drives to the store.
He then buys milk and cookies.
```

milkAndCookies.csv

```
Number of Characters,83
Number of Clauses,3
```

```
Number of Sentences,3
Number of Words,16
```

23 SEP 09

Thesauri, General

Description

The Generalization Thesauri are used to replace possibly confusing concepts with a more standard form (e.g. a text contains United States, USA and U.S. The Generalization Thesauri could have three entries which replace all the original entries with united_states). Creating a good thesaurus requires significant knowledge of the content.

Format of a Thesauri

1. Every line contains a concept found in the text followed by the concept to replace it with. The syntax is **some old concept,some_old_concept**
2. The **original** concept can be one or more words in a row.
3. A **Key** concept **must** be one word.
4. The **original** concept and the **key** concept are separated with a comma.
5. There should not be any space before or after the comma.
6. The Thesaurus is not case sensitive.

Uses for a Generalization Thesauri

Combining multi-word concepts

Peoples names usually consist of two or more individual names like John Smith or Jane Doe.

```
John Smith becomes John_Smith.
```

It is also useful if, after the initial presentation of the full name, a person is referred to by only part of that name. The thesauri would be able to create one concept out of either entry.

```
John Smith becomes John_Smith

John becomes John_Smith.
```

Normalizing abbreviations

Many large companies and organizations are recognized by the abbreviation of their name as well as the name itself.

```
The British Broadcasting Company is routinely known
as the BBC.

The Chief Executive Officer of a company is known as
the CEO.
```

NOTE : Be aware that some ordinary words can be misinterpreted as organizations. One notable example is **WHO - World Health Organization**.

Normalizing contraction

Contractions are used to shorten two concepts into one smaller concept.

```
isn't => is not | I'd => I would | they'll => they
will
```

Expanding these contractions out to their roots allows for creating better Delete Lists.

Correcting typos

When typing people routinely make small spelling errors. Many of these are done when people are not sure of the correct spelling.

```
absense,absence | centruy,century |
manuever,maneuver
```

Or correcting common typing mistakes

```
hte instead of the | chaor instead of chair
```

Globalizing countries

For some countries there are multiple ways to refer to it's name. America, for example, has many ways to reference it's name.

```
US | U.S. | United States | United States of America
| America

Germany | Deutschland (German) | Allemagne (French)
| Niemcy (Polish)
```

Creating a thesauri entry for each of these will reduce the number of concepts in a file while grouping all the same concepts, with variate names, in the same frequency.

Each set can be contained in a separate thesauri and run on a set of texts individually.

Example:

johnInUSA.txt

```
My name is John Smith and I live in the USA.
```

johnInUSAGenThes.csv

```
John Smith,John_Smith
USA,United_States
```

Text after GenThes applied:

```
My name is John_Smith and I live in the
United_States.
```

Thesauri Content Only

Thesauri Content Only creates an output using ONLY the entries found in the thesauri. All other concepts are discarded.

NOTE : When using this option you need to be aware of what is, and is not, in the thesauri.

Example with ThesauriContentOnly not activated

johnInUSA.txt

```
My name is John Smith and I live in the USA.
```

johnInUSAGenThes.csv

```
John Smith,John_Smith
USA,United_States
```

Text after Generalization Thesauri applied:

```
My name is John_Smith and I live in the
United_States.
```

Example using ThesauriContentOnly

TextjohnInUSA.txt

```
My name is John Smith and I live in the USA.
```

johnInUSAGenThes.csv

```
John Smith,John_Smith
USA,United_States
```

Text after Generalization Thesauri applied with ThesauriContentOnly:

```
John_Smith United_States.
```

23 SEP 09

Thesauri, MetaNetwork

Description

Meta-Network (Carley, 2002) associates text-level concepts with Meta-Network categories {agent, resource, knowledge, location, event, group, task, organization, role, action, attributes, when}. One concept might need to be translated into several Meta-Network categories. For example, the concept commander corresponds with the categories agent and knowledge.

The top level of the meta-network ontology is who, what, how, where, why, when. All concepts can be fit to one of these categories.

Meta-Network categories

agent : A person, group, organization, or artificial actor that has information processing capabilities. All "who"s are agents, be they a person in a group, a group within an organization, or the organization itself (e.g. President Barack Obama, the shadowy figure seen outside the building, or the Census bureau). Which sub-category the agents are placed in is left to the user.

knowledge : Information learned such as a school lecture or knowledge learned from experience (e.g. Excellent knowledge of the periodic table or "I know what you did last summer").

resource : Can be either a physical or intangible object. A resource is anything that can be used for the completion of a job. (e.g. One uses a car to drive from point A to point B and money to fund a terrorist organization).

task : A task is part of a set of actions which accomplish a job, problem or assignment. Task is a synonym for activity, although the latter carries a connotation of being possibly longer duration

event : Something that happens, especially something of importance. Events are usually thought of as a public occasions, but they can also be clandestine meetings. The number of agents can range in the thousands or as few as two agents (e.g. Christmas in Times Square or dinner with friends).

organization : A group of agents working together for a common cause (e.g. The Red Cross or the local chess club).

location : An actual physical place. This could be a room in a building, a city, or a country (e.g. Pittsburgh, PA or my living room).

role : An agent's role can be defined as their job for their employer or the part they serve during an event.

action : driving to the mall, eating lunch. Used as a verb.

attribute : Information about the specifics of the agents. These are usually traits that agents have in common, each can be slightly different (e.g. visible traits like hair colour or intangible traits like religious beliefs).

when : Referring to time or circumstances. Can be as broad as a year or as pinpoint as the exact time of a particular day (e.g. Last year or 2:33 PM on March 1st, 2009).

Example:

Let's take two short sentences as an example. It contains **people, places, and things**

dairyFarm.txt

```
Ted runs a dairy farm. He milks the cows, runs the
office, and cleans the barn.
```

dairyFarmDeleteList.txt

There are some unecessary concepts in the text. Using a Delete List will extract the essence of the text. This Delete List is quite short.

```
a, and, in, on, the
```

After applying the delete list, the text appears in the display like this:

```
Ted runs xxx xxx dairy farm. He milks xxx cows, runs
xxx office, xxx cleans xxx barn.
```

Meta-Network Thesaurus:

Now we come to the meta-network thesaurus. This file will define the category for each of the **important** concepts we have in the file.

dairyFarmMeta.csv

```
Ted,agent
runs,task
dairy,resource
farm,location
He,agent
milks,task
cows,resource
office,location
cleans,task
barn,location
```

Examining the File

Generating a DyNetML file in AutoMap prepares it to be examined in ORA.

Description

Thesaurus Content Only is an option used with the **Generalization Thesaurus**. It allows you to select how your results will be display and output.

synopsis-2.txt

```
Synopsis: The Tok'ra plan to kill all the System
Lords. The plan is to infiltrate the summit and
poison the System Lords. But they need a "human" who
speaks gou'ald and that human is Daniel Jackson of
the SGC. He speaks gou'ald. The Tok'ra approach
Daniel, the SGC, and the U.S. Military, with their
plan and he agrees. SG-1 and SG-17 travel with the
Tok'ra to Revenna. After outlining the plan to
Daniel, he is taken by Jacob Carter to the summit
where he is posing as a low ranking gou'ald. O'Neill
stays on Revenna with SG-1 and SG-17. The
assassination plan is proceeding fine until a new
emissary, the gou'ald Osiris, appears. She
recognizes Daniel but stays silent. Daniel and Jacob
both know the assassination of the System Lords
would now cause complications. Meanwhile Revenna is
attacked. O'Neill, Carter, Teal'c, and Elliot help
in the defense of the planet. Daniel escapes the
summit. He joins up with Jacob and they make their
escape back to Revenna intending to rescue O'Neill
and SG-1. Their craft is shot down. Elliot
sacrifices his life in order to allow SG-1 to
escape.
```

synopsis-2GenThes.csv

```
assassination plan,assassination_plan
Carter,Maj_Samantha_Carter
Daniel,Daniel_Jackson
Daniel Jackson,Dr_Daniel_Jackson
Elliot,Lt_Elliot
gou'ald,gou_ald
Jacob,Jacob_Carter
Jacob Carter,Jacob_Carter
low ranking gou'ald,low_ranking_gou_ald
O'Neill,Col_Jack_O_Neill
SG-1,SG1
SG-17,SG17
speaks gou'ald,speak_gou_ald
summit meeting,summit
System Lord,System_Lords
System Lords,System_Lords
Teal'c,Teal_c
the SGC,Stargate_Command
Tok'ra,Tok_ra
```

U.S. Military,US_Military

Thesaurus Content Only - NO : Selecting **NO** will retain all concepts. Thesaurus concepts will be replaced and the entire text will be displayed in the window. Below is an example with the replaced thesaurus entries in bold.

> Synopsis: The **Tok_ra** plan to kill all the **System_Lords**. The plan is to infiltrate the summit and poison the **System_Lords**. But they need a "human" who **speak_gou_ald** and that human is **Dr_Daniel_Jackson** of **Stargate_Command**. He **speak_gou_ald**. The **Tok_ra** approach **Daniel_Jackson**, **Stargate_Command**, and the **US_Military**, with their plan and he agrees. **SG1** and **SG17** travel with the **Tok_ra** to Revenna. After outlining the plan to **Daniel_Jackson**, he is taken by **Jacob_Carter** to the summit where he is posing as a **low_ranking_gou_ald**. **Col_Jack_O_Neill** stays on Revenna with **SG1** and **SG1**7. The **assassination_plan** is proceeding fine until a new emissary, the **gou_ald** Osiris, appears. She recognizes **Daniel_Jackson** but stays silent. **Daniel_Jackson** and **Jacob_Carter** both know the assassination of the **System_Lords** would now cause complications. Meanwhile Revenna is attacked. **Col_Jack_O_Neil**l, **Maj_Samantha_Carter**, **Teal_c**, and **Lt_Elliot** help in the defense of the planet. **Daniel_Jackson** escapes the summit. He joins up with **Jacob_Carter** and they make their escape back to Revenna intending to rescue **Col_Jack_O_Neill** and **SG1**. Their craft is shot down. **Lt_Elliot** sacrifices his life in order to allow **SG1** to escape.

Thesaurus Content Only - YES : Selecting **YES** will eliminate all concepts that do not exist in the thesaurus. The results will depend on a second option choosen.

Thesaurus content only options:

Direct adjacency : All non-thesaurus concepts will be removed form the display and be replaced with a space.

```
: Tok_ra System_Lords. summit System_Lords. ""
speak_gou_ald Dr_Daniel_Jackson Stargate_Command.
speak_gou_ald. Tok_ra Daniel_Jackson,
Stargate_Command, US_Military, . SG1 SG17 Tok_ra .
Daniel_Jackson, Jacob_Carter summit
low_ranking_gou_ald. Col_Jack_O_Neill SG1 SG17.
assassination_plan , gou_ald , . Daniel_Jackson .
Daniel_Jackson Jacob_Carter System_Lords . .
```

```
Col_Jack_O_Neill, Maj_Samantha_Carter, Teal_c,
Lt_Elliot . Daniel_Jackson summit. Jacob_Carter
Col_Jack_O_Neill SG1. . Lt_Elliot SG1 .
```

Rhetorical adjacency : Non-thesaurus concepts are removed from the display but are replaced with a **(xxx)** placeholder. This will show the distance between the thesaurus items.

```
xxx: xxx Tok_ra xxx xxx xxx xxx xxx System_Lords.
xxx xxx xxx xxx xxx xxx summit xxx xxx xxx
System_Lords. xxx xxx xxx xxx "xxx" xxx
speak_gou_ald xxx xxx xxx xxx Dr_Daniel_Jackson xxx
Stargate_Command. xxx speak_gou_ald. xxx Tok_ra xxx
Daniel_Jackson, Stargate_Command, xxx xxx
US_Military, xxx xxx xxx xxx xxx xxx. SG1 xxx SG17
xxx xxx xxx Tok_ra xxx xxx. xxx xxx xxx xxx xxx
Daniel_Jackson, xxx xxx xxx xxx Jacob_Carter xxx xxx
summit xxx xxx xxx xxx xxx xxx low_ranking_gou_ald.
Col_Jack_O_Neill xxx xxx xxx xxx SG1 xxx SG17. xxx
assassination_plan xxx xxx xxx xxx xxx xxx xxx, xxx
gou_ald xxx, xxx. xxx xxx Daniel_Jackson xxx xxx
xxx. Daniel_Jackson xxx Jacob_Carter xxx xxx xxx xxx
xxx xxx System_Lords xxx xxx xxx xxx. xxx xxx xxx
xxx. Col_Jack_O_Neill, Maj_Samantha_Carter, Teal_c,
xxx Lt_Elliot xxx xxx xxx xxx xxx xxx xxx.
Daniel_Jackson xxx xxx summit. xxx xxx xxx xxx
Jacob_Carter xxx xxx xxx xxx xxx xxx xxx xxx xxx xxx
xxx Col_Jack_O_Neill xxx SG1. xxx xxx xxx xxx xxx.
Lt_Elliot xxx xxx xxx xxx xxx xxx xxx SG1 xxx xxx.
```

23 SEP 09

Threshold, Global and Local

Description

Thresholds refine the number of concepts to be included when creating the Union Concept List and the individual Concept List files. As the Threshold number is increased, concepts with frequencies less than the threshold are removed from the file when it is written.

Example Texts

Below are three small text files. They are small for demonstration purposes. As will be seen, even small text repositories can create large Concept List files.

theboy-1.txt : See the boy named Dave. He has two toys. One toy is red and the other toy is blue.

theboy-2.txt : On Monday Dave plays with the blue toy. It's his favorite toy.

theboy-3.txt : On all other days Dave plays with the red toy.

Global Threshold

Using the Global Threshold you can control which concepts will not be included in the Union Concept List. Any concept appearing less than the threshold will not be included in the Union Concept List file that's output.

First create a **Union Concept List** using the unprocessed text files. In large text files this can result in an unwieldy list.

ucl.csv with no pre-processing

```
Words,Frequency,Relative Frequency,Relative
Percentage
all,1,0.2,0.024390243902439025
and,1,0.2,0.024390243902439025
blue,2,0.4,0.04878048780487805
boy,1,0.2,0.024390243902439025
dave,3,0.6,0.07317073170731707
days,1,0.2,0.024390243902439025
favorite,1,0.2,0.024390243902439025
has,1,0.2,0.024390243902439025
he,1,0.2,0.024390243902439025
his,1,0.2,0.024390243902439025
is,2,0.4,0.04878048780487805
it's,1,0.2,0.024390243902439025
monday,1,0.2,0.024390243902439025
named,1,0.2,0.024390243902439025
on,2,0.4,0.04878048780487805
one,1,0.2,0.024390243902439025
other,2,0.4,0.04878048780487805
plays,2,0.4,0.04878048780487805
red,2,0.4,0.04878048780487805
see,1,0.2,0.024390243902439025
the,4,0.8,0.0975609756097561
toy,5,1.0,0.12195121951219512
toys,1,0.2,0.024390243902439025
two,1,0.2,0.024390243902439025
with,2,0.4,0.04878048780487805
Total,41
Mean,1.64
StDev,0.0
```

With these three short files the list is already unwieldy. To decrease the number of concepts, use pre-processing on the raw text using the Delete List, Stemming, and Thresholds

Removing contractions

Notice the text contains the contraction **it's**. In other texts there will probably be many more. Use a thesauri during pre-processing to expand all contractions. This will expand **it's** to **it is** as well any other contractions found in the thesauri file.

Removing plurals

Next we want to combine the concepts of **toy** and **toys**. They both reference the same item and should be counted as the same concept. Run **Stemming** using KSTEM.

Running a Delete List

Use the Concept List Viewer to create a Delete List of unneeded concepts. Then apply this Delete List.

The Revised Union Concept List

Now generate another concept list.

You will find a list of all the **non-deleted concepts**.

```
Words,Frequency,Relative Frequency,Relative
Percentage
all,1,0.16666666666666666,0.030303030303030304
be,2,0.3333333333333333,0.06060606060606061
blue,2,0.3333333333333333,0.06060606060606061
boy,1,0.16666666666666666,0.030303030303030304
dave,3,0.5,0.09090909090909091
day,1,0.16666666666666666,0.030303030303030304
favorite,1,0.16666666666666666,0.030303030303030304
has,1,0.16666666666666666,0.030303030303030304
is,1,0.16666666666666666,0.030303030303030304
it,1,0.16666666666666666,0.030303030303030304
monday,1,0.16666666666666666,0.030303030303030304
name,1,0.16666666666666666,0.030303030303030304
one,1,0.16666666666666666,0.030303030303030304
other,2,0.3333333333333333,0.06060606060606061
play,2,0.3333333333333333,0.06060606060606061
red,2,0.3333333333333333,0.06060606060606061
see,1,0.16666666666666666,0.030303030303030304
toy,6,1.0,0.18181818181818182
```

```
two,1,0.16666666666666666,0.030303030303030304
with,2,0.3333333333333333,0.06060606060606061
Total,33
Mean,1.65
StDev,0.0
```

There's a definite difference between the two lists. Originally there were 25 individual concepts. Now there's a total of 20. Using thresholds will reduce them even further.

Thresholds: Local=1 and Global=2

Now the list can be further refined by setting the **Local and Global threshold** parameters.

First, leave **Local to 1** but change **Global to 2**. This tells AutoMap that a concept must appear a total of two or more times in **all** text files to be included in the Union Concept List.

Create a new concept List.

```
Words,Frequency,Relative Frequency,Relative
Percentage
be,2,0.3333333333333333,0.09523809523809523
blue,2,0.3333333333333333,0.09523809523809523
dave,3,0.5,0.14285714285714285
other,2,0.3333333333333333,0.09523809523809523
play,2,0.3333333333333333,0.09523809523809523
red,2,0.3333333333333333,0.09523809523809523
toy,6,1.0,0.2857142857142857
with,2,0.3333333333333333,0.09523809523809523
Total,21
Mean,2.625
StDev,0.0
```

The origin list contained 25 concepts. After pre-processing it contained 20 concepts. After setting the Global Threshold to 2 it now contains 8 concepts.

Raising the Global threshold to 3 would remove `be, blue, other, play, red, and with` leaving only 2 concepts (dave and toy) in the file.

Local Threshold

The Local Threshold works on individual files. As the threshold is raised, more concepts are removed from the individual concept list files.

Setting the **Local Threshold=2** and the **Global Threshold=1** will remove any concept that appears only once in any of the loaded files.

The results of all three Runs

File	Total number of Concepts in Original File	Concepts written to files using Local Threshold=2
ucl-1.txt	12	2
ucl-2.txt	9	1
ucl-3.txt	8	0

Example of Concept List per Text for ucl-1.txt

```
Words,Frequency,Relative Frequency,Relative
Percentage
be,2,0.6666666666666666,0.4
toy,3,1.0,0.6
Total,5
Mean,2.5
StDev,0.0
```

18 JAN 10

Union

Description

Unioning files/networks is a way of combining two or more files/networks into a single unit. There are multiple ways to union a file or network and each will give differing results.

Union Examples

Let's say for example that the terms **John** and **Mary** both appear in two separate files. Now let's say that in file 1 they are connected three times (frequency=3). And in the second file they are connected nine times (frequency=9).

Minimum

The Minimum union of John and Mary will be the lowest number of connections in either file. In this example a frequency of 3 from file 1 becomes the result.

Maximum

The Maximum union of John and Mary will be the highest number of connections in either file. In this example a frequency of 9 from file 2 becomes the result.

Sum

The Sum union of John and Mary will be a total of all the frequencies added together. In this example file 1 frequency=3 and file 2 frequency=9. The sum of these two is 12.

Average

The Average union of John and Mary will be the sum of the two frequencies divided by the total number of files used. In this example file 1 frequency=3 and file 2 frequency=9. The sum of these two is 12. Next divide this sum (12) by the number of files (2) and the result is 6.

18 JAN 10

Description

The Union Concept List differs from the Concept List in that it considers concepts across all texts currently loaded, rather than only the currently selected text file. The Union Concept List is helpful in finding frequently occurring concepts, including those that, after review, can be added to a Delete List.

The Union Concept List includes:

- The concepts found in all files and the total frequency.

- Related, cumulative frequencies of concepts in all text sets.
- Cumulated unique concepts and total concepts contained in the data set.

NOTE : The number of unique concepts considers each concept only once, whereas the number of total concepts considers repetitions of concepts.

Definitions

Concept : The individual concepts in the file.

POS : Defines the Parts of Speech of each concept

Frequency : Number of times a concept appears in a file.

Relative Frequency : The frequency of any concept divided by the highest value of any frequency

Relative Percentage : The result of adding all of the relative frequency values then dividing a concept's relative frequency by that value.

Example

Start with two (or more) texts.

johnIsAFireman.txt

```
John is a Fireman in lower Manhattan in New York
City. John was there at the Twin Towers on that day
in September.
```

nyc.txt

```
NYC is a city comprised of five boroughs: Manhattan,
Queens, the Bronx, Brooklyn, and Staten Island.
```

A Concept list for each input text:

fireman.csv

```
City,1,0.33333334,single
Fireman,1,0.33333334,single
John,2,0.6666667,single
```

```
Manhattan,1,0.33333334,single
New,1,0.33333334,single
September,1,0.33333334,single
Towers,1,0.33333334,single
Twin,1,0.33333334,single
York,1,0.33333334,single
a,1,0.33333334,single
at,1,0.33333334,single
day,1,0.33333334,single
in,3,1.0,single
is,1,0.33333334,single
lower,1,0.33333334,single
on,1,0.33333334,single
that,1,0.33333334,single
the,1,0.33333334,single
there,1,0.33333334,single
was,1,0.33333334,single
```

nyc.csv

```
Bronx,1,1.0,single
Brooklyn,1,1.0,single
Island,1,1.0,single
Manhattan,1,1.0,single
NYC,1,1.0,single
Queens,1,1.0,single
Staten,1,1.0,single
a,1,1.0,single
and,1,1.0,single
boroughs,1,1.0,single
city,1,1.0,single
comprised,1,1.0,single
five,1,1.0,single
is,1,1.0,single
of,1,1.0,single
the,1,1.0,single
```

A Word list for each input file:

fireman.csv

```
John, is, a, Fireman, in, lower, Manhattan, in, New,
York, City, John, was, there, at, the, Twin, Towers,
on, that, day, in, September
```

nyc.csv

```
NYC, is, a, city, comprised, of, five, boroughs,
Manhattan, Queens, the, Bronx, Brooklyn, and,
Staten, Island
```

A unionConceptList.csv file using both files:

```
concept, frequency, relative_frequency,
relative_percentage
Bronx,1,0.5,0.125
Brooklyn,1,0.5,0.125
Island,1,0.5,0.125
Manhattan,2,1.0,0.25
NYC,1,0.5,0.125
Queens,1,0.5,0.125
Staten,1,0.5,0.125
a,2,1.0,0.25
and,1,0.5,0.125
boroughs,1,0.5,0.125
city,1,0.5,0.125
comprised,1,0.5,0.125
five,1,0.5,0.125
is,2,1.0,0.25
of,1,0.5,0.125
the,2,1.0,0.25
City,1,0.5,0.125
Fireman,1,0.5,0.125
John,2,1.0,0.25
New,1,0.5,0.125
September,1,0.5,0.125
Towers,1,0.5,0.125
Twin,1,0.5,0.125
York,1,0.5,0.125
at,1,0.5,0.125
day,1,0.5,0.125
in,3,1.5,0.375
lower,1,0.5,0.125
on,1,0.5,0.125
that,1,0.5,0.125
there,1,0.5,0.125
was,1,0.5,0.125
```

This Union Concept List can be used as the basis for creating a Delete List or a MetaNetwork Thesauri (Carley, 2002) for all texts loaded.

Using in Excel

A Union Concept List can be sorted in Excel. Open the file in Excel. All the data will appear in a single column. To separate it, first select the column with the data. Then select **Data => Text to Columns** from the menu. In the dialog box select **Delimited** and click Next. Select the check box for **Comma** and click Finish. The data is now in individual columns. To sort the list, highlight the data and select **Data => Sort...**. Select "frequency" under "Sort by" and make sure it is descending. Then select concept under "Then by". Your Union Concept List is sort by frequency.

Window Size

Description

The window size determines the span in which connections will be made. The larger the window size, the more connections within that window.

A conenction is made between each concept within a window. The window will then shift one concept in the direction of the text (for instance, the window shifts right for most Latin-based languages) and create a new window to analyze. This will continue to the end of the text.

Example

cookiesAndMilk.txt

```
I have cookies and milk
```

Window of concepts 1-3 : I have cookies

```
I have, I cookies, have cookies
```

Window of concepts 2-4 : have cookies and

```
have cookies, have and, cookies and
```

Window of concepts 3-5 : cookies and milk

```
cookies and, cookies milk, and milk
```

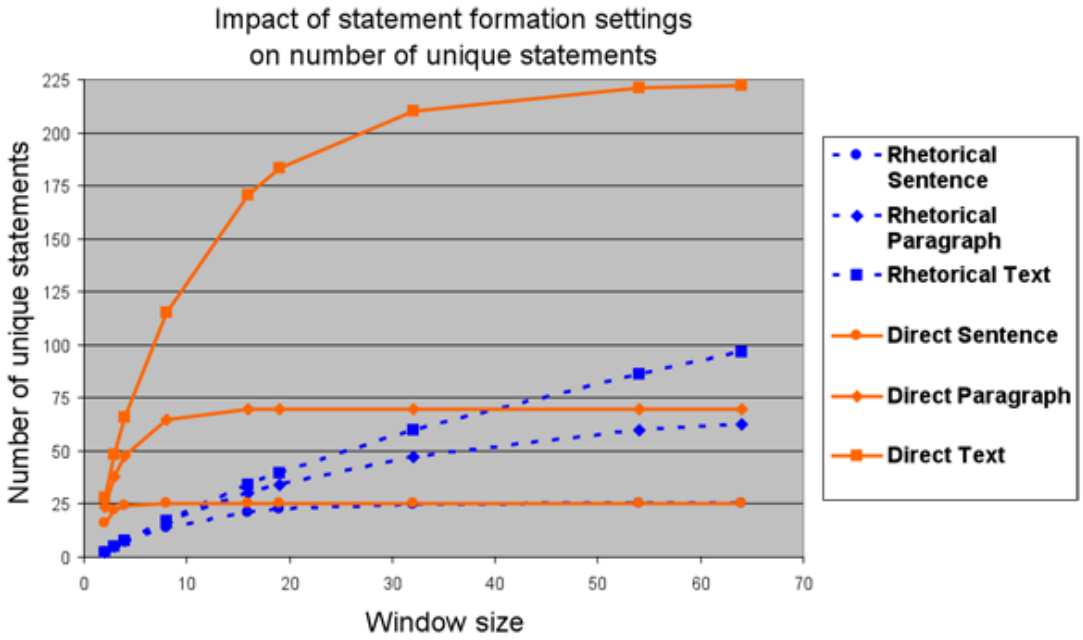
Correct Window Size

Determining a correct window size is important. Choosing too small a window size may result in important links being missed. Too large a window size connects too many concepts, overwhelming important links.

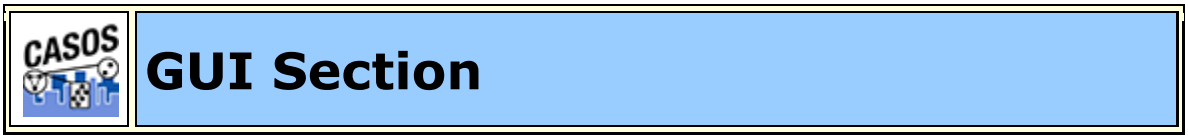
```
Dave likes milk and cookies but John likes
cauliflower
```

The example sentence above contains nine concepts. Manually reviewing this sentence reveals that milk and cookies are associated with Dave and cauliflower is associated with John.

But using a direction of **unidirectional** and a window size of **9** results in cauliflower also being associated with Dave.



18 JAN 10



GUI Section

The AutoMap GUI is a graphic interface for quick visualizations of test files. The section contains pages on:

The GUI

The File Menu

The Edit Menu

The Preprocess Menu

The Generate Menu

The Procedures Menu

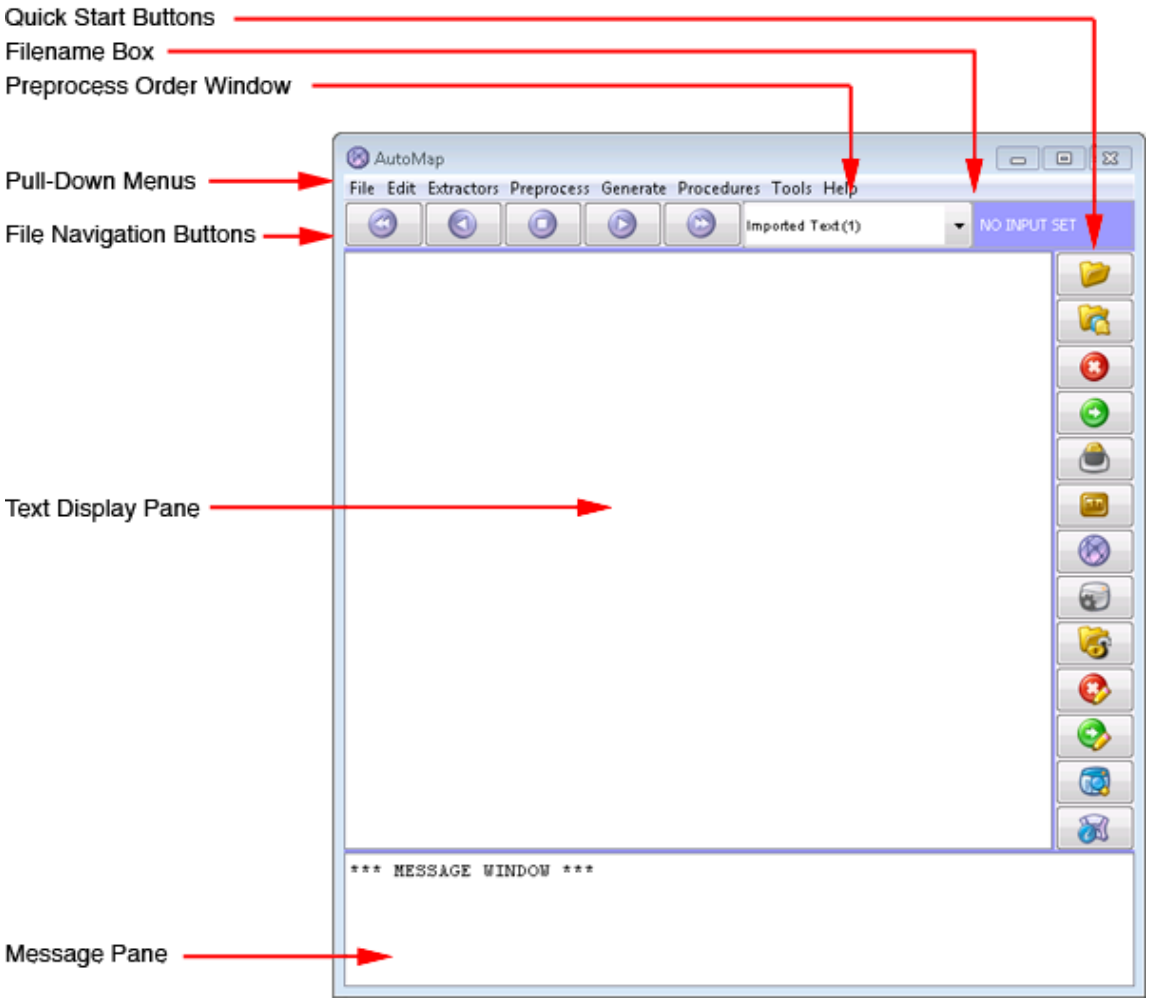
The Tools Menu

The GUI (Graphic User Interface)

Description

The GUI (Graphical User Interface) contains access to AutoMap's features via the menu items and shortcut buttons.

The GUI



The Pull Down Menu

File

Used for loading and saving text files. Can also extract text from other sources (PDFs and the Web).

Edit

Allows the user to change the font of the **Display Window**

Extractors

The File Menu contains functions whose main purpose is working with the files themselves. These functions do not perform any processing or generate any output. They work with raw files to help prepare them for use in AutoMap.

Preprocess

Contains all the preprocessing functions used on your text before generating output. These functions work on original text files only.

Generate

Generates output from preprocessed files. The output is directly related to the work done with preprocessing tools.

Tools

External Tools used by AutoMap to View and Edit output. Tools include Concept List and Semantic List viewers and Delete List and Thesaurus editors.

Help

The Help file and about AutoMap.

File Navigation Buttons

Used to display the files in the main window. The buttons contain from left to right: **First** (lowest numbered file), **Previous** (previous file in sequence), **Goto** (Enter number of specific file), **Next** (next file in sequence), and **Last** (last numbered file)

Keyboard Shortcuts

You can navigate through your loaded files using the cursor keys.

Arrow-Left
Go to the Previous Text File.

Arrow-Right
Fo to the Next Text File.

Arrow-Up
Go to the First Text File

Arrow-Down
Go to the Last Text File

Preprocess Order Window

Contains a running list of the preprocesses performed on the files in the order they were performed. These can be undone one process at a time with the Undo command starting with the last process done. They can not be undone out of order.

Filename Box

Displays the name of the currently active file along with it's ordinal number and the total number of files loaded. Using the File Navigation Buttons will change this and as well as the text displayed in the window. This also displays the total number of files loaded and the order number of the currently displayed text.

Text Display Window

Displays the text for the currently selected file. The name of this file can be found in the Filename Box.

Message Window

Area where AutoMap display the actions taken as well errors encountered. This area is also a place the user can insert notes about the current session. This can be helpful for later reference. You can copy text from the display window or type notes directly.

Quick Launch Buttons

The Quick Start Buttons contain the most frequently used tools.

NOTE : More detailed information about the various functions can be found in the Content and Task sections.

6 NOV 09

File Menu

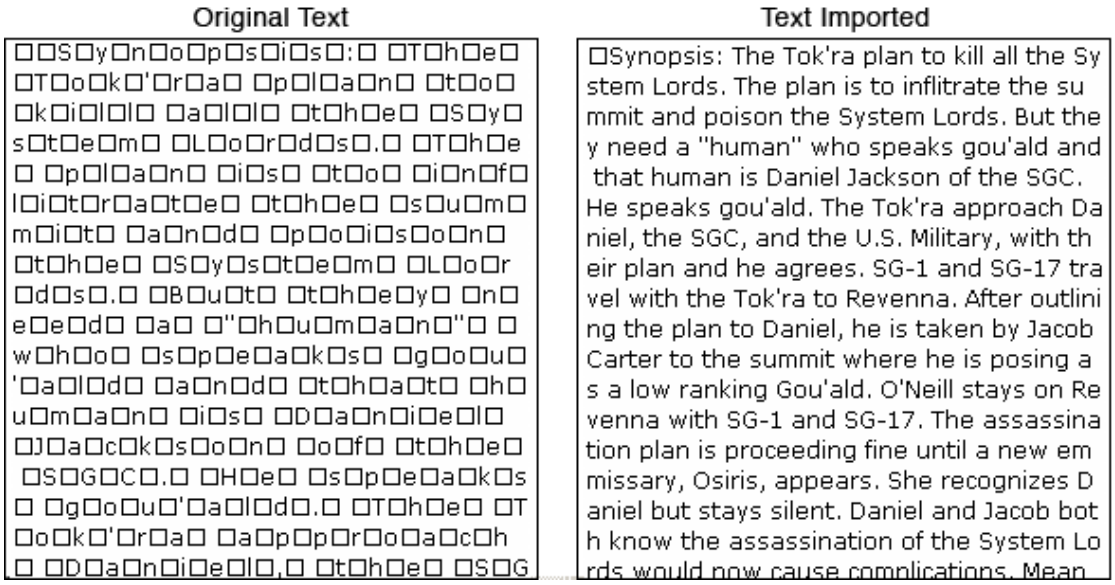
Description

The File Menu contains functions whose main purpose is working with the files themselves. These functions do not perform any processing or generate any output. hey work with raw files to help prepare them for use in AutoMap.

Import Text Files

Allows you to loads one or more files into AutoMap. You can either 1) select an entire directory or 2) select individual files to load. When selecting individual files you can seledt non-contiguous files by holding down the Control key while clicking the files to select. This is the only command which will actually bring text into AutoMap.

If your imported text is in the UTF-8 the first two lines in the Preprocessing Order Window will be identical. But when importing text in other formats.



NOTE : This function only works with text files.

NOTE : When using the script you still have to specify an entire directory.

AutoMap will present a dialog box asking for two parameters.

1. **Select Text Encoding :** This defaults to **Let AutoMap Detect** but you can change this to another encoding if you know the format of your files.
2. **Select Text Direction :** This defaults to **Left->Right, Top->Bottom** but you can change this if you know the direction of your text.

Create New Text File

Creates a blank document in AutoMap which is useful in using copy-and-paste clips from multiple documents. After your new document is complete this file, with any other processing done to it, can be saved.

File Conversion Functions

The functions in the **Conversion Section** help make files more compatible with AutoMap.

File Save Functions

The functions in the **Save section** allow you to save various files from your work in AutoMap.

Exit AutoMap

When you exit, AutoMap will ask if you want to save your preferences. Remember, there are two sets of preferences: **User and System**. This includes the directory you visited last, the options you used when you created a metanetwork (directionaly, window, etc.), your font choices, and others. These will be restored the next time you start AutoMap.

NOTE : It will not save the state of loaded files after exiting.

After saving the preferences it will close all files and exit.

26 OCT 11

File Menu-Conversions

Extract Standard Files

Uses standard files included with AutoMap and lets you save them in a different location so they can viewed or edited without affecting the original files.

Extract SVN File

Given an address to a file in a specified SVN repository it will extract the file from the repository and rename it to the file specified by the user. You may either enter your user name and password as part of the dialog prompt or script arguments, but they are only optional. Once the program is run, if a user name

and password were not provided, the executable program itself will prompt the user to enter his or her user name and his or her password.

 *Check File Encoding*

AutoMap will ask you to navigate to a file. The encoding of the selected file will be displayed in the message window.

 *Convert File to UTF-8*

AutoMap will ask you to selecte a file. This file will be converted it to the UTF-8 format. AutoMap will only correctly convert text tiles. If you try to convert non-text files and it will convert them **incorrectly**.

This function works a single file at a time.

 *Compare Text Files*

Compares two text files and tells you what percentage the files have in common with each other.

 *Flatten Directory*

This operation will copy all files from a hierarchy of directories into a single directory. It will rename files if needed in the case of two files from different subdiretories having the same name. AutoMap requires all input files to be in a single directory.

 *Group Rename Files*

This operation will create a new directory of files with the names of the file to be renamed based on a mapping provided by the user via a CSV file of original file name and requested new file name. If a file name is not mentioned, the original file will be copied with no change to the file name.

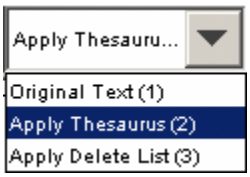
File Menu-Save

Save Preprocessed Text Files

By default this saves all text files at the highest level of preprocessing (e.g. the last process shown in the Preprocess Order Window. This procedure can be done any number of times during processing.

NOTE : If you need to keep a copy of a previously saved set of processed text files you need to either move the first set of files to a new directory or rename the files before you save a new set of files.

You can also save a set of processed text files from any executed set inshown in the Preprocess Order Window. Highlight the step at which to save the text (see below) and then select this function.



Save Intermediary Text File

Works almost identically to **Save Preprocesed Text Files** except it inserts the **Bell** character at the end of each sentence. This assists in allowing AutoMap in finding the end of sentences.

Save Script File

After performing all your preprocessing steps on your test file you can save the entire procedure as a script file **(e.g. a file ending in .config)**. AutoMap will write out the tags based on the list of preprocesses and with the parameters you set.

*See **Tools => Script Runner** for more information.*

Save Message Window Log

During an AutoMap session all of your requests will be reflected in the message window.

NOTE : This window is also **user alterable** meaning you can insert notes regarding this session. After completing your work you can save this file for future reference.

7 JAN 11

Edit Menu

Description

Show MetaNetwork Text Tagging

Creates a MetaNetwork (Carley, 2002) List for each loaded file based on a selected MetwNetwork Thesaurus. AutoMap will ask you to specify a target directory for the lists it creates. Will tag any concept found in the MetaNetwork Thesaurus. All others are tagged as **UNKNOWN**.

Show Part of Speech Tagging

Parts of Speech assigns a single best **Part of Speech**, such as noun, verb, or preposition, to **every word in a text**. While many words can be unambiguously associated with one tag, (e.g. computer with noun), other words can match multiple tags, depending on the context that they appear in.

```
...
Roman,JJ
citizens,NNS
wandering,VBG
the,DT
...
```

NOTE : When finished reviewing the information you should **Undo** this item or else any further processing will be done using the results obtained. You mostly want to see the results without have AutoMap continue processing with these results.

Filter Words

Enter a word or words you want to filter your Imported Text through. Upon finishing AutoMap will display only those words for each text.

NOTE : When finished reviewing the information you should **Undo** this item or else any further processing will be done using the results obtained. You mostly want to see the results without have AutoMap continue processing with these results.

 ***Hide Known Entries***

This routine will prompt for a master thesauri and then hide known thesauri entries in the main text window.

NOTE : When finished reviewing the information you should **Undo** this item or else any further processing will be done using the results obtained. You mostly want to see the results without have AutoMap continue processing with these results.

Preference Menu : Where you can change various parameters affectin how AutoMap functions.

26 OCT 11

 ***Set Font***

Allows the user to change the font used in the display window. This is important because if you view a file in the wrong font it will **NOT** display properly. Some characters will be displayed improperly while others may be displayed as empty boxes.

NOTE : It is important to note that **Font** and **Encoding** are not the same thing.

*See **Content => Encoding** for more information.*

 *Set Font Size*

Allows the user to change the default font size used in the display window. Choose a size between 8 and 48 points.

 *Show User Preferences*

Displays your current settings in a dialog box. These are either the last saved settings (If you used the Save User Preferences) or the setting for the current session (If you have never saved preferences or reset them).

> These preferences include: **Current Font and Size, Current Working Directory, Window Size, Direction Preference, Stop Type Preference, Stop Value Preference, Viewer Preference, Thesaurus Sort Preference, File Union Preference, and Pop-Up Preference.**

 *Save User Preferences*

Saves the preferences listed above. The next time work is begun AutoMap will used the preferences in saved file.

 *Reset User Preferences*

Resets all previously saved user settings.

```
font=Ariel
cwd=whatever your computer is set for
Viewers=Ask Each Time
```

 *Viewer Launch Preference*

After certain processes are run on text files you may, or may not, want to view them. You can choose from the options above how you want AutoMap to handle this situation.

 *Thesaurus Sort Preference*

The Sort Thesaurus sorts by the number of words in a key_concept. You can select how you want AutoMap to handle the sorting of your thesaurus before processing.

Union Preference

Some AutoMap functions have an option to create a union file after individual files are processed. You can shoose how you want AutoMap to handle creating, or not creating, a union file.

Pop-up Preference

The Pop-Up preference differs from the previous controls as it has only two choices : **Always Do It** and **Never Do It**

Color Preference

The dialog box will display the **Current Window Color** and the **Current Font Color**. Below that it display an example of the two colors. You can click on either box to bring up the color picker and change either one. When you are satisfied with your color choices, click **Save**.

Storage Preferences

Allows you to specify where AutoMap will store all output. Since AutoMap saves all output in distinctively named folders and will never overwrite any previous output.

Temporary Workspace Preferences

Allows you to specify where AutoMap will store all temporary files it uses. The default is to use your C: drive. Changing this is useful if your C: drive is too small for the amount of data you are processing.

Heap Size Preferences

Allows you to specify the memory size to run external program tools. If you have any issues of running out of memory increasing the Heap Size will solve that. 32-bit systems have a

limit 1-GigaByte and the 64-bit systems have a limit of 265 GigaBytes.

26 OCT 11

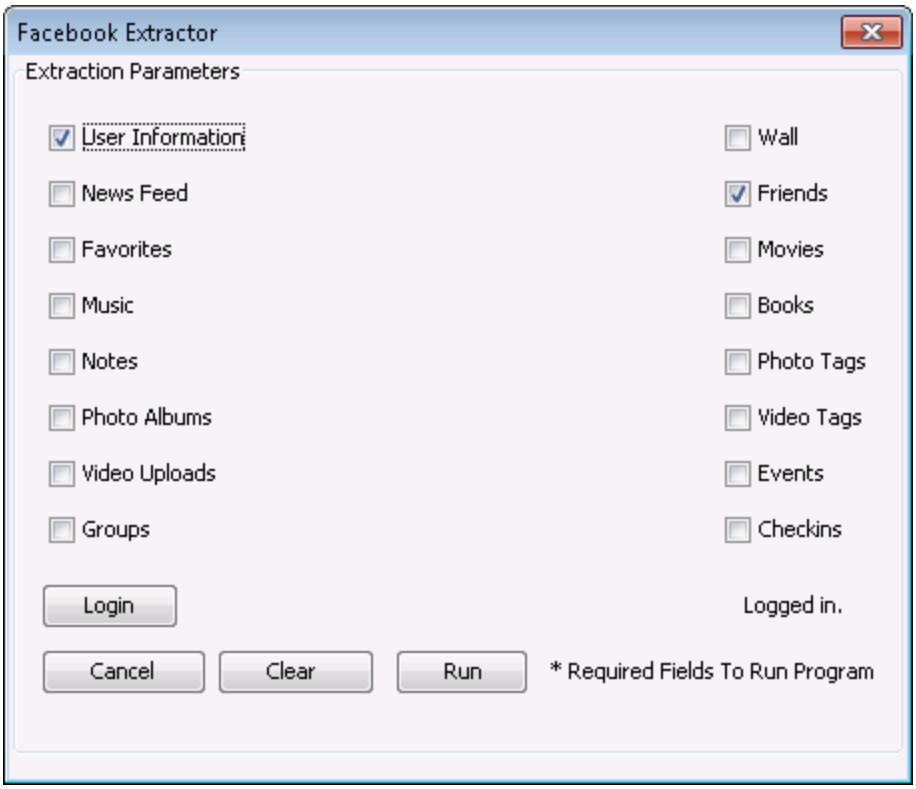
CEMap

This menu is a group of functions used to extract information from the web. This is done with various tools depending on the web site to download from.

NOTE : Remember that when it asks for a URL that you must put in the **http://** or other necessary protocol.

Blogs Extractor : Enter the URL of the blog. Be aware that this is not just the URL of the blog but the URL of the feed for the blog. [i.e. for Blogspot.com you would attach **/feeds** to the end of the blog URL.]

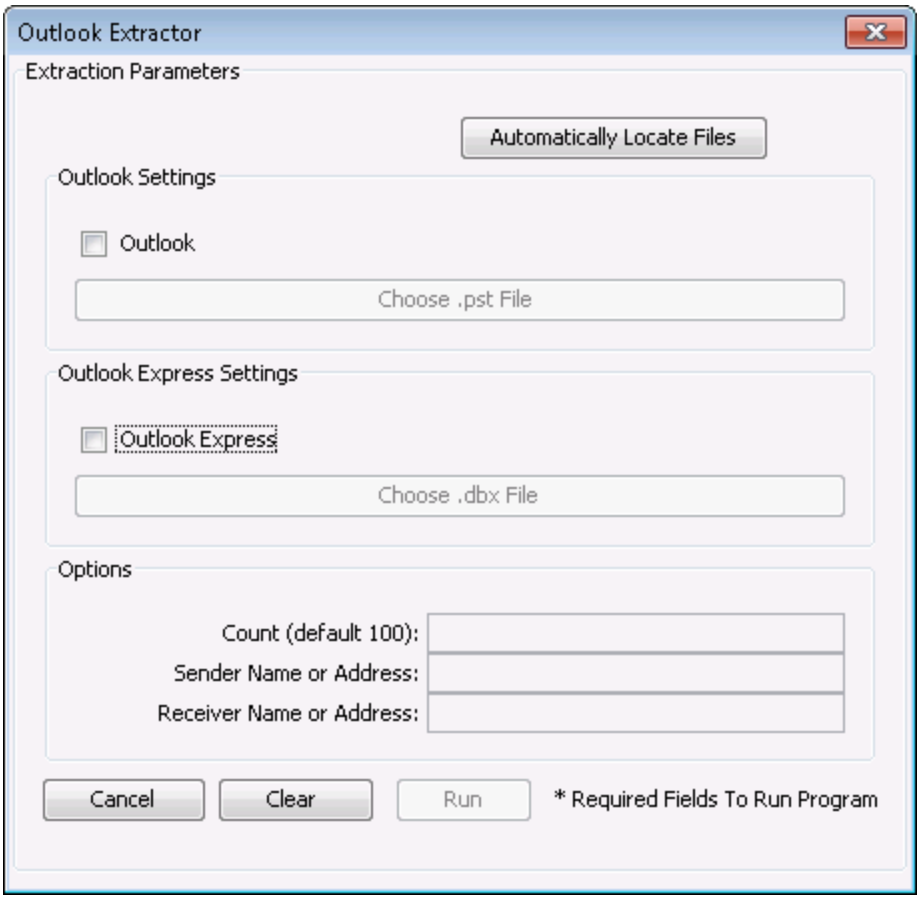
Facebook : Used to extract information from a Facebook account for which you have the username and password. Place a check mark next to the information to extract.



Mail : This contains three options:

MBox : This is a file on your computer which contains all the emails for an account. They are downloaded from the server, saved into the file, then deleted from the server.

Outlook :



POP3/IMAP :

Extractors Menu

Description

The File Menu contains functions whose main purpose is working with the files themselves. These functions do not perform any processing or generate any output. They work with raw files to help prepare them for use in AutoMap.

Extract Web Pages to Text Files

To extract text from a web site you **point AutoMap** to the web site and it will extract everything that page touches **that is on the same site**. It won't go beyond that main site (e.g. no external links). AutoMap will then ask you for an output directory. Make sure the output directory is empty.

All files will be renamed taking a web sites hierarchical structure and creating a flat file (a list of files with no hierarchical structure of folders). The renaming occurs so as to not over-write different files with the same filename. Each file will receive a unique identifying name. It may not be logical to the person but it helps maintain order.

There will also be a file written to the directory called **Index**. This file will have **no extension**. It has no .txt extension so it will never **accidentally** be processed in AutoMap. Index (AutoMap's mapping file) takes the big long web filenames and shows what it has been transformed into. After extracting a web site it is often good to run Deduplicate Files. In many cases there might be files with duplicate content but different filenames.

NOTE : AutoMap creates many small files from each section of the web page. An average web page could generate hundreds of text files.

Replace HTML Symbols

Converts HTML code [**i.e. &#33, ©, ½**] and converts then to single concepts [**!, ©, ½**] .

NOTE : This does not remove/replace HTML tags.

Transform Documents into Text Files

The types of documents that can be transformed by AutoMap are: **Adobe PDF, Microsoft Word 2003 (.doc), Microsoft Excel 2003 (.xls), Microsoft PowerPoint (.ppt) and HTML files**. This function will extract text from these files **if it is available**.

It will only read one type at a time. You will be asked for the type from the dropdown menu. Then you will be asked to navigate to the directory where the files are located. If you want to convert multiple types of files you will need to perform this function multiple times, once for each type.

NOTE : When attempting to convert PDF files be aware that some PDFs contain **images** of the text only. AutoMap can not

read this text. Also be aware that some **non-Adobe** programs create PDFs which may create incompatible PDFs which AutoMap can't extract.

Copy Selected Text Files

Allows you to copy selected files to another directory. AutoMap will ask you to define a **filter** to detect which files to copy. You can use one of two wild card symbols. The **(*)** symbol takes the place of multiple characters in a filename. (e.g. file1.txt, file20,txt, and file315.txt would all be copied using the filter file*.txt). The **(>)** character is used to replace a single character (e.g. file1.txt and file2.txt would be found using file?.txt but file200.txt would not)>

NOTE : You must specify an output directory **other than** your original input directory

Deduplicate Text Files

Locates duplicate files in a directory. It works on the principle of the file's content, not the filename.

AutoMap will ask you for the directory to check then ask for a directory to write it's output. Two directories will be created: log (containing a text file of it's actions) and removed texts (the text files which are duplicates).

NOTE : The files in the original directory remain unaffected.

If you choose to deduplicate by a **Percentage Difference Threshold** which is the percentage the files need to be alike. 5% means only 5% of the text in the files need to match. 90% means 90% of the text in the files need to match.

Strip File Headers

First select a directory containing text file(s). Make sure it's not empty or it will error out. Next you select an output directory to save the new file(s). Click [**OK**] when asked for **Enter an optional phrase to remove.**. Now enter the number of paragraphs to strip from the file(s). A new set of files will be written to the output directory.

NOTE : A paragraph is considered any set of line(s).

NOTE : A blank line is also considered a paragraph for this purpose.

Split Text Files

Splits all text files in selected directory using the number of paragraphs input. (i.e. a file with 15 paragraphs and an input of three could create five files of three paragraphs apiece).

NOTE : A blank line is considered a paragraph because the function counts line returns as paragraph markers.

Recombine Text Files

Used to recombine files previously split using the **Split Text Files** function.

Semi structured CSV

Takes column entries with unstructured text and saves each row as a separate file. First you specify the csv file to use. Next give it the columns which contain unstructured data. For this use the column labels in Excel (i.e. A, B, C...)

AutoMap will create one file for each row of the file with only the columns specified.

For this example I specify columns A and C as unstructured.

A	B	C
alpha	aaaaa	lolcat
bravo	bbbbb	fizzbin
charlie	ccccc	oicurmt
delta	ddddd	jelly babies
echo	eeeee	titanic
foxtrot	fffff	why

And the six files written would be:

```
File 1: alpha,lolcat
File 2: brave, fizzbin
File 3: charlie, oicurmt
File 4: delta, jelly babies
File 5: echo, titanic
File 6: foxtrot, why
```

Preprocessing Menu

Description

Following is a short description of the preprocessing functions in AutoMap3. These functions serve to prepare files to deliver output by reducing unneeded and unwanted concepts.

More detailed information can be found in the Content section as well as the individual tutorials and lessons.

 ***Undo Last Step***

Undo removes the **last** Preprocessing done to the text. This is done one step at a time. To remove multiple preprocessing steps you must perform multiple undos.

 ***Redo All Steps***

Reprocesses all Preprocessing steps. Useful if new text files are added or a support file has been altered.

 ***Preform All Cleaning***

Contains functions for basic text clean up. Remove Extra Spaces, Fix Common Typos, Convert British to American Spelling, Expand Common Contractions, Expand Common Abbreviations, and Replace HTML Symbols.

Text Cleaning Functions Descriptions

 ***Preform All Preparation***

Contains functions to further prepare text. Pronoun Resolution, Remove American Letters, NGram Conversion, Remove Pronouns, Remove Noise Verbs, Remove Presitions, and Remove All Noise Words.

Text Preparation Functions Descriptions

Text Refinement

Contains the functions to finalize text. Remove Num bers, Remove Punctuation, Remove User Symbol, Remove Single Symbol, Remove Symbols, Convert to Lowercase, Convert to Uppercase, Apply Stemming, Apply Delete List, Apply Generatlization Thesauri.

Text Refinement Functions Descriptions

26 OCT 11

Text Cleaning Menu

Description

Preform All Cleaning

Performs all steps below in one step.

Text Cleaning Sub-Menu

Remove Extra Spaces

Removes all cases of multiple white spaces and replaces them with a single space. Regardless of the initial number of spaces, the end result will be one white space.

*See **Content > Remove White Space** for more information*

Convert British to American Spelling

Converts British spellings **i.e. humour** to American spellings **to humor**.

Fix Common Typos

Fixes the most common typos in the English language.

Expand Common Contractions

Changes common contractions **can't, I'm, won't** to separate words **can not, I am, will not**.

Expand Common Abbreviations

A thesauri which finds common abbreviations [**i.e. U.S.A. OR NYC**] and converts then to single concepts [**United_States or New_York_City**].

29 MAR 11

Preprocessing Menu

Description

Preform All Preparation

Performs all steps below in one step.

Text Preparation Sub-Menu

Pronoun Resolution

Find pronouns in text and resolves to whom the pronoun is referring.

```
Mike went to the store. He bought milk.

Mike went to the store. Mike bought milk.
```

NOTE : AM3 will always look backwards in the text to resolve a pronoun - not forwards.

Remove Single Letters

Removes any single letter, either lower- or uppercase

NGram Conversion

Creates single concepts from multi-word ngrams by replacing the space between the words with an underscore.. **i.e. cut off becomes cut_off**. These phrases are two or more words in length.

Remove Pronouns

Removes all pronouns such as [**he, she, and it**].

Remove Noise Verbs

Removes all noise verbs such as [**is, am, and was**].

Remove Prepositions

Removes all prepositions such as [**on, but, and till**].

Definition : A preposition links nouns, pronouns and phrases to other words in a sentence. A preposition usually indicates the temporal, spatial or logical relationship of its object to the rest of the sentence.

Remove All Noise Words

Performs all of the above functions.

Remove Day and Month Words

Removes all day and month words such as [**Monday, Tue, and July**].

Remove Numbers as Words

Removes all number words such as [**one, thirty, and hundred**].

Remove Possessive Form

Removes the possessive form of words and converts them to their non-possessive form.

Remove Complete Numbers

Removes numbers that make up the entire concept such as [**123, 6, and 8988**] but not [**F22 or C3PO**].

Convert Hyphenated Words to N-grams

This routine will replace all hyphenated words with their n-gram form replacing the hyphen with an underscore.

Reconcile Full Names

This routine will attempt to reconcile a name (identified as a proper noun) with a previously identified multi-word name (identified as a sequence of proper nouns) such as firstname lastname. The single name will be replaced by the multi-word name.

26 JAN 11

Refinement Menu

Description

Remove Numbers

Removing numbers will remove not only numbers as individual concepts but also removes numbers embedded within concepts. The option is to remove completely or replace with a white space.

*See **Content > Remove Numbers** for more information*

Remove Punctuation

The Remove Punctuation function removes the following punctuation from the text: **.,:;' "()!?-**. You will have the option to remove them completely or replace them with a white space.

*See **Content > Remove Punctuation** for more information*

Remove User Symbols

If you only want to remove a subset set of the symbols you can create a txt file with only those symbols. The **Remove User Symbols** function will ask for the location of that file and AutoMap leave the remaining symbols in your files.

Remove Single Symbol

Automap asks for **one symbol** to remove from the text file(s).

*See **Content > Remove Symbols** for more information*

Remove Symbols

The list of symbols that are removed:
~`@#$%^&*_+={}[]\|/<>. You will have the option to remove them completely or replace them with a white space.

Convert to Lowercase

Convert to Lowercase changes all text to **lowercase**.

Convert to Uppercase

Convert to Uppercase changes all text to **UPPERCASE**.

*See **See Content > Format Case** for more information*

Apply Stemming

Stemming removes suffixes from words. This assists in counting similar concepts in the singular and plural forms (e.g. plane and planes would normally be considered two terms). After stemming planes becomes plane and the two concepts are counted together. Two Stemmers are available, **K-Stem and Porter**.

*See **See Content > Stemming** for more information*

Apply Delete List

A Delete List is a list of concepts to be removed from a text files. It is primarily used to reduce the number unnecessary concepts. By reducing the number of concepts being processed run times are decreased and semantic networks are easier to understand. This also helps in the creation of a semantic network in reducing the number of superficial nodes in ORA.

*See **Content > Delete List** for more information*

Apply Generalization Thesauri

The Generalization Thesauri are used to replace possibly confusing concepts with a more standard form (e.g. a text contains United States, USA and U.S. The Generalization Thesauri could have three entries which replace all the original entries with united_states). Creating a good thesaurus requires significant knowledge of the content.

*See **Content > Thesauri, General** for more information*

Merge Hyphenated Words at Line Ends

If a hyphenated word is at the end of line it is followed by a **end-of-line character**. Removing hypens would result in half the word on one line while the second half of the word started the next line. This routine removes both the hyphen and **end-of-line character** character then combines them into one word.

29 JUL 11

Generate Menu

Description

The following are short descriptions of the functions from Generate Pull Down menu. These functions generate output from preprocessed files.

When you run any of the generate functions AutoMap will create a new folder for the results. The folder will begin with the preprocess function end with a number (e.g. MetaNetwork1, MetaNetwork2...). AutoMap will find the last number in the series and increment the number by one. If no folder exists then AutoMap will create a new folder starting with 1.

 ***Text Properties***

Outputs information regarding the currently loaded files. AutoMap writes one file for each file currently loaded containing.

```
Number of Characters,14369
Number of Clauses,325
Number of Sentences,167
Number of Words,2451
```

*See **Content => Text Properties** for more information.*

 ***Named Entities***

Named-Entity Recognition allows you to retrieve proper names, numerals, and abbreviations from texts.

*See **Content => Named Entity** for more information.*

 ***Data Extraction***

The Feature Selection creates a list of concepts of money, dates, phone numbers and times.

*See **Content => Feature Extraction** for more information.*

Part of Speech Sub-Menu :

Concept Lists Sub-Menu :

Semantic Networks Sub-Menu :

MetaNetworks Sub-Menu :

Thesaurus Suggestion Sub-Menu :

Generalization Thesaurus Sub-Menu :

8 SEP 11

Generate-Parts Of Speech

Part of Speech Tagging

Parts of Speech assigns a single best **Part of Speech**, such as noun, verb, or preposition, to **every word in a text**. While many words can be unambiguously associated with one tag, (e.g. computer with noun), other words can match multiple tags, depending on the context that they appear in.

AutoMap will ask you how you want to save your files. First Automap will ask if you want **Standard** (the entire list of tags) or **Aggregation** (a consolidated list) Parts of Speech tagging. Second you will be asked to save them in the **CSV** or **TXT** format.

```
...
Roman,JJ
citizens,NNS
wandering,VBG
the,DT
...
```

*See **Content => Parts of Speech** for more information.*

POS Attribute File

Similar to the above function but if there are multiple occurances of the same concept it will assign **the best possible Part of Speech** to a concept.

```
...
battlefield,NN
volumnius,PRP
benefit,NN
angrily,CD
...
```

Verb Extraction

Complies of list of all actions (verbs) in the specified file.

Noun Extraction

Complies of list of all nouns and in the specified file.

26 OCT 11

Generate-Concept Lists

Concept List (Per Text)

Generates a Concept List for all loaded files. The list contains a concept's frequency (number of times it occurred in a file), relative frequency (a concept's frequency in relationship to the total number of concepts). A Concept List can be refined using other functions such as a Delete List (to remove unnecessary concepts) and Generalization Thesaurus (to combine n-grams into single concepts).

concept	pos	frequency	relative frequency within text	gram type	number of texts	meta
Antony	NNP VBN	4	0.14814815	single	1	UNKNOWN
Brutus	EX IN JJ NN NNP PRP VBN	16	0.5925926	single	1	UNKNOWN
Caesar	DT NNP VB VBN	27	1.0	single	1	UNKNOWN

*See **Content => Concept List** for more information.*

Concept List (Union Only)

The Union Concept List differs from the Concept List in that it considers concepts across all texts currently loaded, rather than only the currently selected text file. The Union Concept List is helpful in finding frequently occurring concepts, and after review, can be determined as concepts that can be added to the Delete List.

*See **Content => Union Concept List** for more information.*

Concept List with MetaNetwork (Carley, 2002) Tags

Creates a Concept List which lists a MetaNetwork category if applicable.

Concept Network DyNetML (Per Text)

Creates a separate DyNetML file of concepts for each text file loaded. These files are directly usable in ORA.

Concept Network DyNetML (Union Only)

Creates one DyNetML file of the concepts in all text files loaded. This is a union file of all concepts. This file is directly usable in ORA.

NOTE : Both Concept Network functions create DyNetML files with one NodeClass and no Networks. Making the connections is up to you after importing the file into ORA.

NOTE : Leading and trailing hyphens are removed before generating Concept Lists and Semantic Lists but hyphens in the middle of two words are not (e.g. **because-- something** removes the double hyphens but in the concept **t-shirt** the hyphen would not be removed).

Keywords in Context

A list will be created so every concept in a file along with the concepts which both precede it and following it.

```
concept,left,right
Two,,tribunes
tribunes,Two,Flavius
```

```
Flavius,tribunes,and
and,Flavius,Murellus
...
```

NOTE : The first entry **Two,,tribunes** contains a blank entry for **left** as it's the first word in the text and has nothing to the left. A similar entry will be found at the end with a blank in the column **right**.

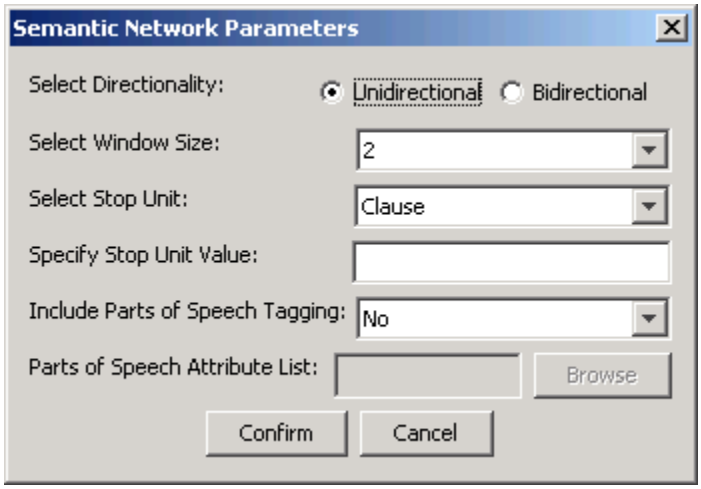
7 JAN 11

Generate-Semantic Networks

Semantic Network DyNetML (Per Text)

Semantic networks are knowledge representation schemes involving nodes and links between nodes. It is a way of representing relationships between concepts. The nodes represent concepts and the links represent relations between nodes. The links are directed and labeled; thus, a semantic network is a directed graph. Semantic Networks created can be displayed in ORA.

NOTE : Use All Words As Window Size : When generating a Meta Networks, Semantic Networks or Semantic Lists, the user now has the option of specifying whether or not they want to use all words in a sentence as the window size. Using this option will automatically set the stop unit to "sentence".



Semantic Network DyNetML (Union Only)

Creates union file of all DyNetML files in one directory. Before running this make sure that only the DyNetML files you want to union reside in the directory choosen.

<p

*See >**Content => Semantic Network** for more information.*

Semantic (Co-Reference) List

Semantic Lists contain pairs of concepts found in an individual file and their frequency in the chosen text file(s).

*See **Content => Semantic List** for more information.*

NOTE : Leading and trailing hyphens are removed before generating Concept Lists and Semantic Lists but hyphens in the middle of two words are not (e.g. **because-- something** removes the double hyphens but in the concept **t-shirt** the hyphen would not be removed).

Noun Phrase to Modifiers List

Creates a .csv file containing what AutoMap believes to be noun phrases using the format **concept, concept, frequency, metaOntology, metaName**.

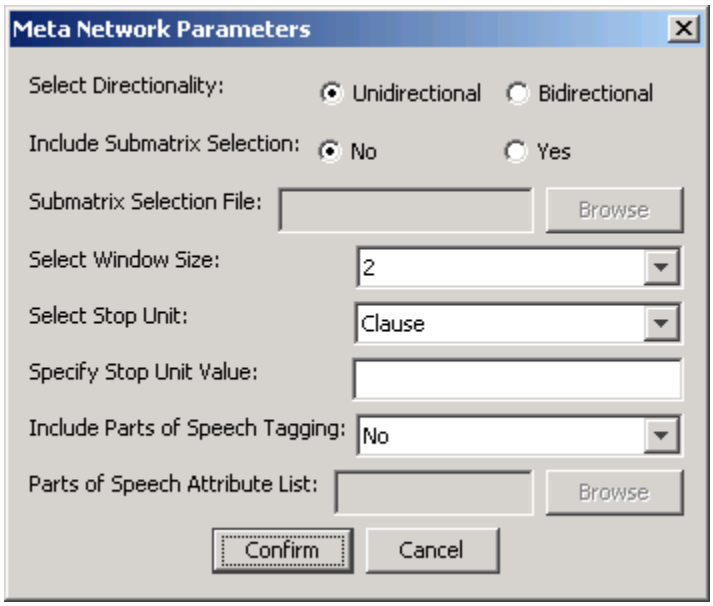
Concept to Source List

Creates a csv file containing a list of concepts and their origin file and frequency, **concept, source, frequency, metaOntology, metaName**.

Modal Rule List

MetaNetwork DyNetML (Per Text)

Assigns MetaNetwork (Carley, 2002) categories to the concepts in a file. This is used to create a DyNetML file used in ORA.



Select Directionality sets whether AutoMap will search only forward in the text or will perform a search both forward and barckward in the text. **Select Window Size** sets the farthest distance from a word to another for a possible connection. **Select Stop Unit** contains Word, Clause, Sentence, Paragraph, or All.

NOTE : The panel contains defaults for all parameters except the **Specify Stop Unit Value**

NOTE : Use All Words As Window Size : When generating a Meta Networks, Semantic Networks or Semantic Lists, the user now has the option of specifying whether or not they want to use all words in a sentence as the window size. Using this option will automatically set the stop unit to "sentence".

MetaNetwork DyNetML (Union Only)

Creates union file of all MetaNetwork (Carley, 2002) DyNetML files in one directory. Before running this make sure that only the MetaNetwork (Carley, 2002) files you want to union reside in the directory choosen.

NOTE : The Union type is a sum type.

MetaNetwork Text Tagging

Tags any entry found in the text files which exists in the selected General Thesaurus using the format **concept/part of speech**.

Suggested Name Thesauri

Creates a file with the following attributes: conceptFrom, conceptTo, frequency, relative_frequency-across_texts, relative_percentage-across_texts, number_of_texts, metaOntology, metaName.

And example taken from the unprocessed Julius Caesar files:

Concept From	concept To	frequncy	relative frequency across texts	relative percentage across texts	number of texts	meta Ontology	Meta Name
Julius Caesar	Julius_Caesar	1	0.0192	0.0030	1	agent	
Brutus	Brutus	4	0.0769	0.0122	2	agent	
Cassius	Cassius	13	0.2500	0.0396	4	agent	

Suggested Uncatagorized Thesauri

conceptTo	pos	metaOntology	conceptFrom
That	DT	knowledge	
angrily	CD		
parade	NN	resource	
wrongly	RB	knowledge	

26 OCT 11

Generate-Thesaurus Suggestion

Suggest MetaNetwork Thesauri (Unigrams Only)

Automatically estimates mapping from text words from the highest level of pre-processing to the categories contained in the Meta-Network. The technology used is a probabilistic model based on a conditional random fields estimation. Suggested thesaurus is a starting point.

```
1In,resource
1On,resource
Cicero,agent
sons,agent
streets,location
Brutuss,agent
women,agent
prisoner,agent
4Portia,resource
masses,agent
...,...
```

A MetaNetwork (Carley, 2002) Thesaurus associates concepts with the following metanetwork (Carley, 2002) categories: **Agent, Knowledge, Resource, Task, Event, Organization, Location, Action, Role, Attribute, and a user-defined categories**.

NOTE : The more the text is modified the less accurate the CRF generator will be.

*See **Content => MetaNetwork** for more information.*

NOTE : The following five functions create a Suggested MetaNetwork Thesaurus along with cleaned files that use the newly created Thesaurus.

OneMode : Assigns a single metaOntology to each entry

MultiMode : Assigns all matching metaOntologies to each entry

Suggest Entity Thesaurus (OneMode, includes multiword expressions)

Creates a thesaurus with **conceptFrom, conceptTo, metaOntology, POS, frequency**. Then applies this thesaurus to files currently loaded into AutoMap.

Suggest Entity Thesaurus (MultiMode, includes multiword expressions)

Creates a thesaurus with **conceptFrom,conceptTo,metaOntology,POS,frequency**. Then applies this thesaurus to files currently loaded into AutoMap.

Suggest MetaNetwork Thesaurus, Categories and Specificity, recommended default (Multimode, includes multiword expressions)

Creates a thesaurus with **conceptFrom, conceptTo, metaOntology, metaType, POS, frequency**. Then applies this thesaurus to files currently loaded into AutoMap.

Suggest MetaNetwork Thesaurus, Categories and Subtype (Multimode, includes multiword expressions)

Creates a thesaurus with **conceptFrom, conceptTo, metaOntology, metaType, POS, frequency**. Then applies this thesaurus to files currently loaded into AutoMap.

Suggest MetaNetwork Thesaurus, Categories, Specificit, and Subtype (Multimode, includes multiword expressions)

Creates a thesaurus with **conceptFrom, conceptTo, metaOntology, metaType, metaname, POS, frequency**. Then applies this thesaurus to files currently loaded into AutoMap.

Decision Support Wizard

A simple chart to assist you in determining the correct method to Thesaurus Creation.

Generate-Generalization Thesauri

BiGrams

BiGrams are two adjacent concepts in the same sentence. If a Delete List is run previous to detecting bi-grams then the concepts in the Delete List are ignored. Multiple Delete Lists can be used with a set of files.

NOTE : The two concepts of a bigram can not cross a sentence or paragraph boundary

*See **Content => BiGrams** for more information.*

Context-Sensitive Stemming Thesauri

Takes concepts down to their base forms. It makes a thesauri for users to evaluate and run.

- It depluralizes nouns, such as "boys" to "boy".
- It detenses verbs, such as "ran" to "run".

Non-Context Stemming Thesauri

Creates a thesaurus with the information **conceptFrom, conceptTo, metaOntology, metaName, frequency, POS**. Will stem entries to their bases. [**i.e. believing > believe, gives > give, and enraged > enrage**]

Identify Possible Acronyms

Given a directory with text files in it, Identify Possible Acronyms will scan through the text files of the given directory and compile a list of possible acronyms. It identifies acronyms as a series of letters in all upper-case, so not all acronyms will be completely accurate; hence why it is a list of possible acronyms. The program takes an input directory and an output directory, and creates a single CSV named >s[pan>possibleAcronyms.csv. It has two column headers: one for the acronym, and the other for the frequency in which acronym appears across all text files. So an example file would contain such:

```
CONCEPT,FREQUENCY
US,3
WHO,5
USA,10
```

Suggest NGram List

Computes a list of possible NGrams from a given set of input text files. It creates a list of **bigrams, trigrams, quadgrams and quintgrams** and then unions those lists together. Invalid ngrams specified by a set of rules the program follows are removed from this union list before it is written out to the file specified by the user.

Positive Thesaurus

A Positive Thesaurus takes every concept in the text and defines it as itself. This can be used as the start in building a Generalization Thesaurus.

NOTE : This function is **case specific** meaning if the concepts **He** and **he** both appear in the text they will both appear in the newly created thesaurus.

fido.txt

```
John has a dog named Fido
```

Positive Thesaurus

```
John,John
has,has
a,a
dog,dog
named,named
Fido,Fido
```

Procedures

Description

This group of functions work on files other than the currently loaded text files.

Validate Script

Determines whether a script is valid to run in AutoMap.

Basic Model Wizard

Master Thesauri Sub-Menu :

Concept List Sub-Menu :

Thesaurus Sub-Menu :

Delete List Sub-Menu :

DyNetML Sub-Menu :

20 JAN 11

Procedures-Master Thesauri

IMPORTANT NOTE : It is necessary to make sure the headers in a Master Thesaurus contain the proper headers before using them.

conceptFrom	conceptTo	metaOntology	metaName

They do not necessarily have to be in that order but they need to be those exact names.

Master Thesauri Merge

Click the Original Master Thesauri [**Browse**] button and select a file to change. Click the Change Thesauri [**Browse**] button to select a second Thesauri file. Underneath use the radio buttons to select the type of File this is. Click the Output Master Thesauri Directory [**Browse**] button and navigate to the location to save the new file.

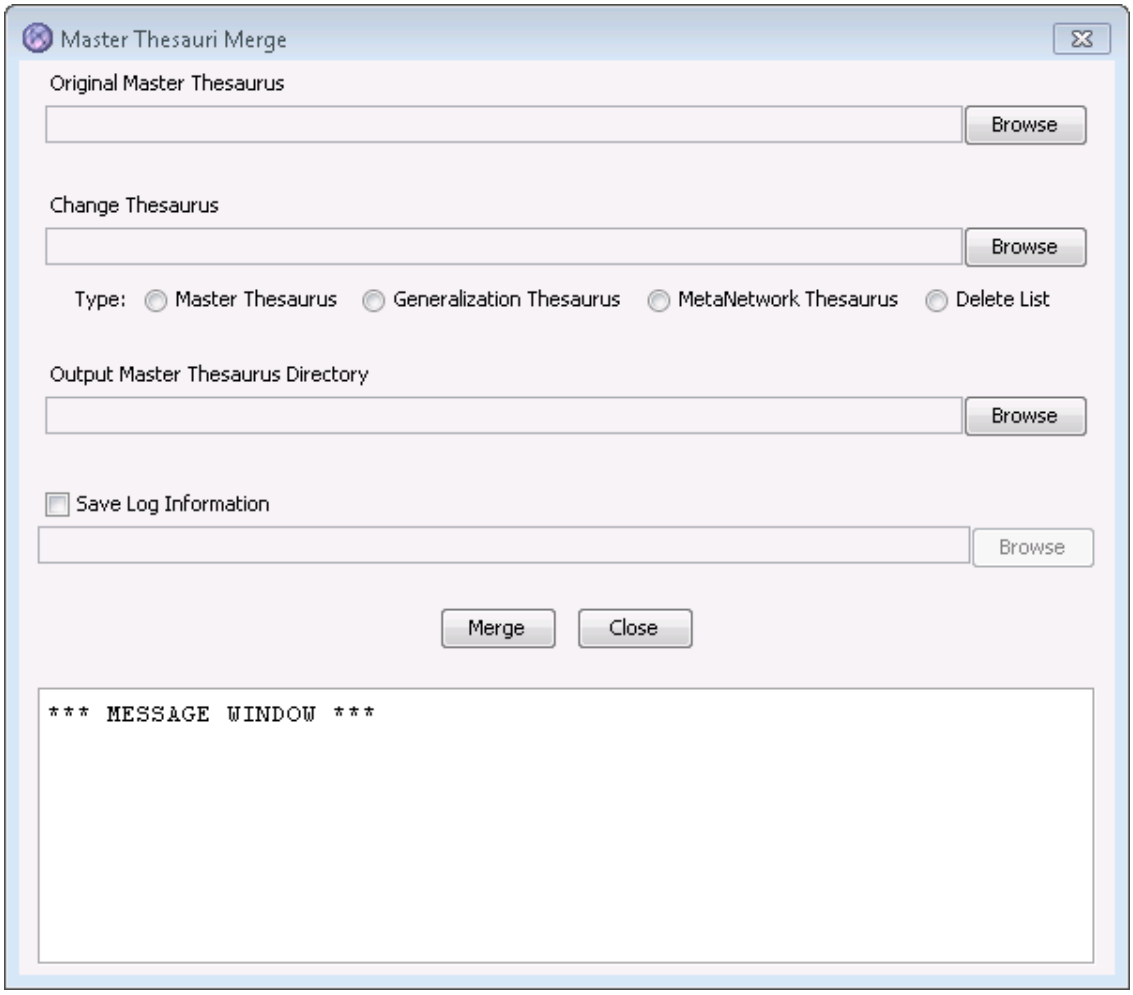
Convert UTF Entries to ASCII Entries

Converts the **unreadable** characters into their readable equivalent. This works on the thesauri files. If there is no equivalent for a character on the line, it is written out to a rejects file. You will be asked for [**1**] the file to convert; [**2**] the name of the file to write converted characters; and [**3**] the name of the file for leftover characters which can not be converted.

NOTE : All three files require the **.csv** extension.

NOTE : If a check mark is placed in the **Save Log Information** you can use the Click the [**Browse**] button to select a location to save this file.

When finished click [**Merge**] to create the newly merged file.



Convert Review to Master

Approximate Closeness of Names

This procedure takes as input a master thesaurus and a threshold percentage of how similar you want names to be. What is output is a master thesaurus with similar names being generalized to another similar name. For example, given the following input:

```
conceptFrom,conceptTo,metaOntology,metaName
Dan,Dan,agent,casos
Dave,Dave,agent,casos
Frank,Frank,agent,casos
France,France,location,country
Mike,Mike,agent,casos
Pike,Pike,resource,weapon
```

The output we get when using a word distance threshold of 80% is:

```
"conceptFrom","conceptTo","metaOntology","metaName"
"Frank","France","agent",""
"Mike","Pike","agent",""
```

NOTE : A **Word Distance Threshold** of 95% gives a good balance.

Extract Thesaurus Attributes

Take apart an input master thesaurus and break it down into multiple files based on an entry's ontological value. For example, all of the agent types in a sample input thesaurus will show up in the **inputName_agents.csv** file generated by this procedure. The same can be said for organizations, locations, tasks, resources, and so on. A meta thesaurus, a delete list and a generalization thesaurus are also created from this process.

Identify Bad Characters in Thesaurus Entries

This will go through a thesaurus and identify each potentially problematic character on each line. The output will list the line number of the thesaurus first, and then all of the problem characters found, then each entry where that problem character occurs. Here is the list of characters we identify as bad:

' ~ = [ ] \\ ; ` . / ! @ # $ % ^ & * ( ) + { } | : < > ? \t

Identify Thesauri Noncategorized Entries

Takes as an input a Master Thesauri and will display in the message window all entries which have no metaOntology listed for it. This list will appear in the Message Window. The

information in this window can be to a file via **File > Save Message Window Log** for use in other programs.

> **NOTE :** Does not work on Generalization Thsauri. The must first be converted to the Master Thesauri format.

 *Derole Thesauri Entries*

Takes as input a thesaurus and outputs a thesaurus, both in master format. Will attempt to find **roles** in both the **conceptFrom and conceptTo** columns of the thesaurus and add de-roled terms to the thesaurus. An attribute file is also output from the program, that contains a list of what roles are mapped to which concepts.

> The entry **President Barack Obama** would add two concepts to the thesaurus: 1) **Barack Obama** and 2) **President Barack Obama**.

 *Apply Thesauri as Delete List*

Takes three different Master Thesauri as arguments: an input thesaurus, an output thesaurus and a delete thesaurus. The Delete Thesaurus is treated as a Delete List and is applied to the Input Thesaurus.

> **NOTE :** The Master Format is required for all arguments.

 *Apply Ontology Rules*

Takes an Input Thesaurus and outputs a Thesaurus with modified Meta Ontology values. The program reads in a encrypted rules file, which contains rules to reclassify a concept's meta ontology value based on patterns.

> **Example :** A concept that contains **Press Release** at the end of it would be reclassified to a task. The current rules file is encrypted distributed within the installers.

 *Remove Noise Patterns*

Takes an Input Thesaurus and outputs a Thesaurus with special patterns stripped out of the list of concepts.

NOTE : Examples would be: letter-_letter or -_letter or _letter.

Separate Number Terms from Thesauri

Takes an Input Thesaurus and outputs two different Thesauri. The first is a thesaurus with all number concepts stripped out of it, except number concepts that are potentially locations. The second thesaurus is a thesaurus of only the number concepts that have been removed from the input thesaurus.

Resolve Names

This is a program that takes a master thesaurus as input and outputs a master thesaurus. The program will scan through the conceptFrom column of the input thesaurus and find entries that have a meta ontology value of agent. The program will then compile a list of possible names to resolve to, only storing the longest possible term for each name. Lastly, the program will scan through the list of agents in the thesaurus once more and -- if the entire term is a part of the full name listed -- the program will set that term's conceptTo column as the full name.

This :

```
Mark Godwin,Mark_Godwin,agent,person
Mark,Mark,agent,
Godwin,Godwin,agent,
```

Will be resolved to :

```
Mark Godwin,Mark_Godwin,agent,person
Mark,Mark_Godwin,agent,
Godwin,Mark_Godwin,agent,
```

NOTE : This feature has been implemented into the script, the AutoMap GUI and the Script Runner GUI. It has also become a part of the deletion process and will automatically be run when NameThesaurusRevision is called.

Remove Leading Article

Takes a Master Thesaurus as input and outputs a Master Thesaurus. It will scan through the **conceptTo** column of the

input thesaurus and find entries that begin with either **a, an or the**. Those prefixes are then removed.

Start with :

```
The John Smith
Corporation,The_John_Smith_Corporation,organization,
```

Will change to :

```
The John Smith
Corporation,John_Smith_Corporation,organization,
```

 ***Split Compound Thesauris Entries***

Takes a Master Thesaurus as input and outputs a Master Thesaurus. Scan through the **conceptFrom** column of the input thesuarus and find entries that contain **and, or, and the bullet character (\u2022)**. It then takes that concept apart and adds each separated concept to the thesaurus as a term, with its meta ontology value being derived from the compound concept.

NOTE : The only exception to this is if the program encounters an organization with **and** in it. If there is one **and**, then the concept is left together. Otherwise, it is separated.

Example :

```
Blue Cross and Blue
    Shield,Blue_Cross_and_Blue_Shield,organization,
Andy and Brian and Charlie and Donna and Ed and
    Frank,Andy_and_Brian_and_Charlie_and_Donna_and_Ed_
    and_Frank,agent
•_eggs_•_milk_•_bread_•_cinnamon_powder_•_cheese,resou
    rce,
```

Will change to :

```
Blue Cross and Blue
Shield,Blue_Cross_and_Blue_Shield,organization
Dan,Dan,agent,
Mike,Mike,agent,
Frank,Frank,agent,
Dave,Dave,agent,
Jessica,Jessica,agent,
Bradley,Bradley,agent,
eggs,eggs,resource
milk,milk,resource
```

```
cinnamon powder,cinnamon_powder,resource,
cheese,cheese,resource,
```

Remove Date Entries

Looks in Concept List and any entry that matchs the date format is removed.

Revise Name Thesaurus

Takes a Master Thesaurus as input and outputs a master thesaurus. Scans through the **conceptFrom column** of the input thesaurus and find entries that have a meta ontology value of agent. Then it compiles a list of possible names to resolve to, only storing the longest possible term for each name. Lastly, it will scan through the list of agents in the thesaurus once more. And if the entire term is a part of the full name listed the program will set that term's **conceptTo column** as the full name.

Example :

```
Mark Godwin,Mark_Godwin,agent,person
Mark,Mark,agent,
Godwin,Godwin,agent,
```

Will resolve to :

```
Mark Godwin,Mark_Godwin,agent,person
Mark,Mark_Godwin,agent,
Godwin,Mark_Godwin,agent,
```

Split to Individual Ontologies

08 SEP 11

Procedures-Concept List

Concept List Procedures

Union Concept List Together

With this function you can join **any** concept lists into a Union Concept List file, even if they are from different textsets. Place all the concept lists you want to union into an empty directory. Then navigate this function to that directory. It will create a union of all the files in a newly created sub-directory called **union**.

 *Concept List Trimmer*

First you select **Trim by file percentage** or **Trim by frequency percentage**. AutoMap will as for a Concept File to trim then a name for the new file. Next you will be asked for either a percentage or frequency to trim the file.

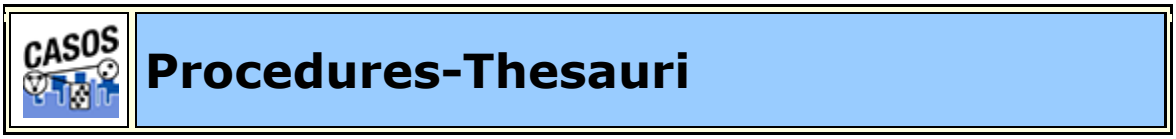
 *Apply Delete List to Concept List*

Allows you to chose a Delete List to apply to a selected Concept List

 *Remove Integers from Concept List*

Removes all numbers from a Concept List

Convert Concept to Review



Thesaurus Procedures

 *Sort Thesaurus*

In certain situations it is important to have your thesaurus sorted from longest to shortest before using it in the preprocess section. Entries with the most number of words are floated to the top of the list

johnSmithDairyFarm.csv - Unsorted

```
John Smith,John_Smith
cow,animal
dairy farm,dairy_farm
pig,animal
The United States of America,the_USA
```

```
chicken,animal
Jane Doe,Jane_Doe
```

johnSmithDairyFarm.csv - Sorted

```
The United States of America,the_USA
John Smith,John_Smith
dairy farm,dairy_farm
Jane Doe,Jane_Doe
cow,animal
pig,animal
chicken,animal
```

The United States of America with five words floats to the top. This is followed by the three entries **John Smith, dairy farm, and Jane Doe** each with two words. It finishes with three entries **cow, pig, and chicken** each with one word.

NOTE : If your thesaurus has duplicate entries (e.g. "John,John_Doe" and "John,John_Smith") a warning will appear in the message window. **Warning: Duplicate entries found in thesaurus for "John"**.

Merge Generalization Thesaurus

Combine multiple Generalization Thesauri into one file. AutoMap allows you to select individual files from a directory.

NOTE : When giving the new file a name remember to also add the .csv extension.

NOTE : If a concept exists in two thesauri but have different key_concept values then both will be included in the merge.

NOTE : This procedure only supports the older file format. If you have a Master Format file then use the Master Thesaurus Merge.

Apply Stemming to Thesauri File

Takes a thesaurus file and creates new entries if a concept requires stemming. If multiple entries are stemmed to the same root and they have different key_concepts then new entries will be added for each one.

drive.csv

```
drove,alpha
driven,bravo
```

Thesaurus after Stemming

```
drove,alpha
driven,bravo
drive,alpha
drive,bravo
```

 ***Apply a Delete List to a Thesaurus***

You can use a Delete List to trim a Thesaurus.

 ***Check Thesaurus for Missing Entries***

Checks a thesauri to find any where either line is blank.

 ***Check Thesaurus for Duplicate Entries***

Checks if there are two entries referencing the same item. This is deteremined by the original concept.

 ***Check Thesaurus for Circular Logic***

Sometimes, when creating a generalization thesauri, a concept is accidentally listed as both something to be replaced and something to replace another concept. For example:

```
United States,US
cow,animal
US,United_States_of_America
```

In this case, all instances of "United States" will first be changed to "US" and then to "United_States_of_America". The Circular Logic Test alerts the user of this inefficiency.

 ***Check Thesaurus for Conflicting Entries***

Will alert you if two or more Thesaurus entries are directed to replace the same concept.

The following four procedures convert files between formats as the names state.

Convert Master Thesauri to Generalization Thesauri

Convert Generalization Thesauri to Master Thesaurus

Convert Master Thesauri to Meta-Network Thesaurus

Convert Meta-Network Thesauri to Master Thesaurus

7 JAN 11

Procedures-Delete Lists

Delete List Procedures

Apply Stemming to DeleteList File

Either the **K-Stem** or the **Porter** stemmer can be applied to a delete list, each with clightly different results.

deleteListToStem.txt

original list	K-Stem	Porter
drives	drives	drives
	drive	drive
wanted	wanted	wanted
	want	want
financial	financial	financial
		financi
motivation	motivation	motivation
		motiv

Merge Delete Lists

Combine multiple Delete Lists into one file. AutoMap allows you to select individual files from a directory.

NOTE : When giving the new file a name remember to also add the .txt extension.

NOTE : Wildcards are not supported when designating file names.

NOTE : This procedure only supports the older file format. If you have a Master Format file then use the Master Thesaurus Merge.

Convert Master Thesauri to Delete List

Takes a Delete List in the Master Thesauri format and converts it to a Standard Delete List.

Delete List - Master format

```
"conceptFrom","conceptTo","metaOntology","metaName"
"a","a","#",
"about","about","#",
"actually","actually","#",
"after","after","#",
"all","all","#",
```

Delete List - Standard format

```
a
about
actually
after
all
```

Convert Delete List to Master Thesauri

Performs the complementary function of the preceeding item.

21 APR 11

Procedures-DyNetML

DyNetML Procedures

Union DyNetML

Will combine all DyNetML files within a single directory into a new DyNetML file.

 *Add Attributes*

Used to add a attributes to a DyNetML file before importing them into ORA. The format of the attribute file is:

header row : `NodeID, Attribute Name, ...., Attribute Name`
data row : `Node Name, Attribute value, ...., Attribute value`
`Additional rows or data`

This will create an attribute column in the DyNetML underwhich all the values for identified nodes will be displayed.

NOTE : If the DyNetML file does not contain a particular node_ID then no information for that node_ID will be added to the file.

Example

```
NodeID,color,shape
alpha,red,circle
beta,green,square
charlie,blue,triangle
```

 *Belief Enhancement*

 *Relocate Source Location in DyNetML*

Changes the source reference in a DyN etML file.

 *Add Icon Reference to DyNetML*

 *Pairwise Union*

Takes as input two DyNetML files which need to be in separate folders. It then creates a third DyNetML file which combines the nodes and links of the two source files.

NOTE : The names of both source files needs to be identical.

Remove Semantic Networks

Removes the Semantic Network from a Meta-Network. This is done to reduce the size of the DyNetML if the analyst is only interested in the Meta-Network and has no use for the semantic net.

Infer Kinship Links

Builds DyNetML files for all XML files located in the source directory. Results are based on frequency and proximity.

Tools Menu

Description

This section contains external tools for working with files outside what is loaded into the GUI. Any work done here is independent of the files that are loaded.

Delete List Editor

Used to modify existing Delete Lists and create new lists. It can compare two Delete Lists and display the difference between them.

*See **Tools > Delete List Editor** for more information.*

Thesauri Editor

Used to modify existing thesauri files by adding or subtracting pairs of concepts. You can also compare two thesauri files and display the difference between them.

*See **Tools > Thesauri Editor** for more information.*

Attribute Editor

*See **Tools > Attribute Editor** for more information.*

 ***Concept List Viewer***

Used to view concept lists or compare two concept lists then display the differences. You can also create Delete Lists from any list currently displayed.

*See **Tools > Concept List Viewer** for more information.*

 ***Table Viewer***

Used to open up any **.csv** file. The major difference between this and the other tools it can compare tables with different amounts of columns.

*See **Tools > Table Viewer** for more information.*

 ***XML Viewer***

The XML viewer can examine any XML file which includes both Semantic Network files and your DyNetML files. Each file will display it's structure and the individual properties of the nodes and networks.

*See **Tools > XML Viewer** for more information.*

 ***Tagged Text Viewer***

A viewer that can be used to view text files which have been tagged with **Parts-of-Speech** or **MetaNetwork** tags.

*See **Tools > Tagged Text Viewer** for more information.*

 ***Script Runner***

Script Runner allows you to run an AutoMap script without opening a Command Window.

*See **Tools > Script Runner** for more information.*

 ***Location Distillation***

 ***Text Partitioner***

Divides a file into the number of highlighted sections created. Highlighting alternates colors as each new section done.

*See **Tools > Text Partitioner** for more information.*

27 OCT 11

Tools

Description

This section contains descriptions of the tools contained in AutoMap. The Tools include:

Delete List Editor

Thesaurus Editor

Attribute Editor

Concept List Viewer

Table Viewer

XML Viewer

Tagged Text Viewer

Script Runner

Location Distillation

Text Partitioner

General Notes about Tools

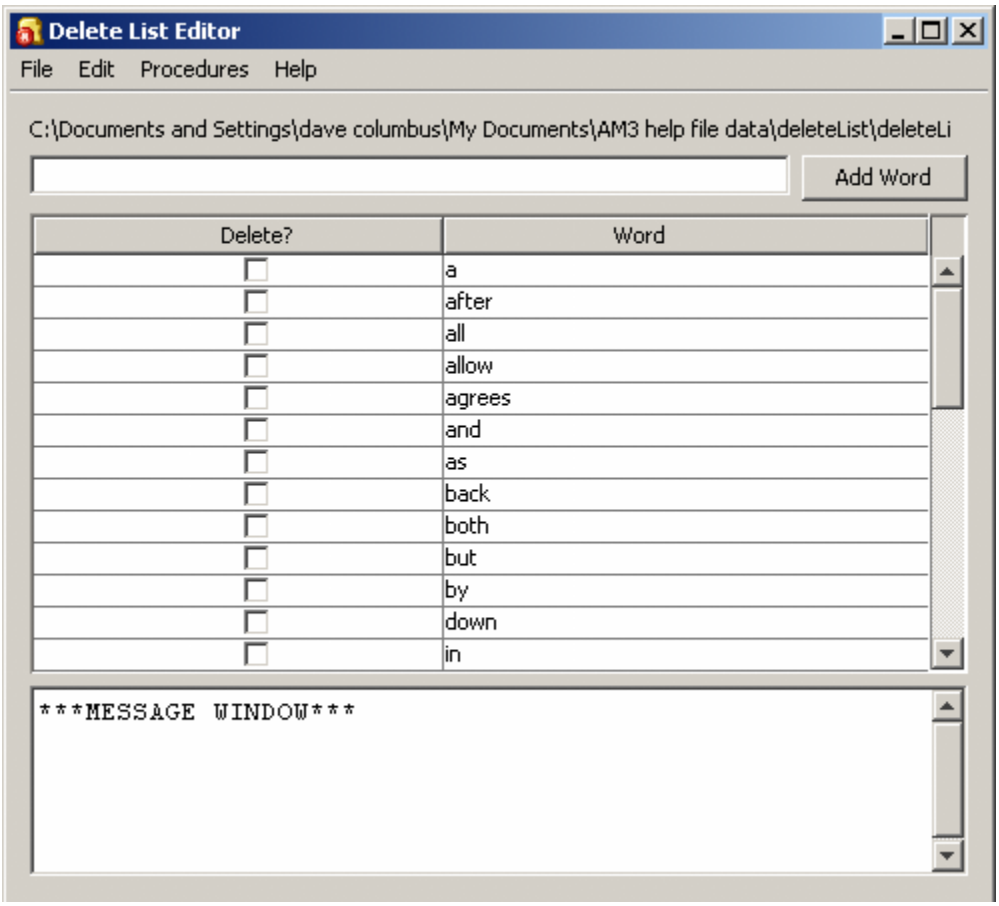
- When running comparisons AutoMap will display details about the comparison in the Message Log Window. This can include some or all of the following: **Lines added, Lines deleted, Lines modified**. More information can be had on the **Compare Colors Page**
- When saving files in any tool the location where the file is saved will be displayed in the Message Pane.

6 NOV 09

Delete List Editor

Description

The **Delete List Editor** can modify existing Delete Lists or create new Delete Lists.



GUI

- **Adding New Words:** You type a word to add in the textbox then click the **[Add Word]** button. The new word will be added to the list.
- **The Message Window :** Displays message from AutoMap and records all your actions while in the editor.

NOTE : No concepts are added or deleted until you actually save the file.

Sorting

To sort the list click on any of the headers. AutoMap will sort the entire list by the clicked header in an **ascending order**. Clicking that same header again will sort the list in a **descending order**. Clicking a different header will once again sort in an **ascending order**.

NOTE : The small triangle to the right of the header will tell you which header is used for sorting and whether it's in ascending **upward facing arrow** or descending **downward facing arrow** order.

Pull-Down Menus

The File Menu

Open File : Allows the user to select a Delete List to load into Editor. The file should be in the format of **one concept per line**.

NOTE : If you load a regular text file then each paragraph will be displayed as a single concept in the viewer.

Save : Saves the Delete List the the same location it was imported from. The location of the saved file is displayed in the message window.

Save as... : Saves a Delete List but allows the user to give the file a **new name and save it to a different new directory** than the original.

Save Message Log : Saves the message log from the Delete List window.

Convert File to UTF-8 : Attempts to convert an input file into the UTF-8 format.

Exit : Exits the Delete LIst Editor and returns to the Main GUI.

The Edit Menu

Compare : Compares a second Delete List to the currently loaded Delete List.

Add Terms from Concept List : Asks user to select a Concept List which will be added to the currently loaded Thesaurus.

Add Terms from NGram : Asks user to select an NGram List which will be added to the currently loaded Delete List.

Add Stemmed Terms : Adds stemmed words to the currently open Delete List. The User will be asked whether to use the Porter Stemmer or the K-Stemmer.

Select All : Selects every concept by placeing a check mark in every box in the **Delete?** column.

Select None : Unselects every concept by removing the check marks from every box in the **Delete?** column.

Remove Selected : Removes the concepts which contained a check mark in the Delete? column. The original file remains unaffected.

Identify Possible Misspelling : Highlights in yellow concepts AutoMap may consider misspelled. Hovering over

these concepts will give a list of alternatives.

Find : You can search for an exact word or use the **(*)** as a wildcard which substitutes for one or more characters.

NOTE : Searching for **t*e** would find **the, there, and theatre** (if all three were in your list.

Reset Colors : Clears the color backgrounds from all cells.

NOTE : The colors are cleared but any extra cells from the compared file remain on screen. To do a new comparision open a new file.

The Procedures Menu

The functions in this pull-down menu do not affect the currently loaded Thesaurus. They are identical to the functions that can be found in the Main GUI.

Apply Stemming to Delete List : You are asked to select a stemmer to apply (Porter Stemmer or K-Stem). All newly stemmed words will be added to the Delete List on screen. You need to use one of the **Save options** to keep this new list.

Merge Delete Lists : Allows you to select two or more Delete Lists and combine them into one. AutoMap will then prompt you to save the new Delete List with a new name and location.

19 APR 11

Thesauri Editor

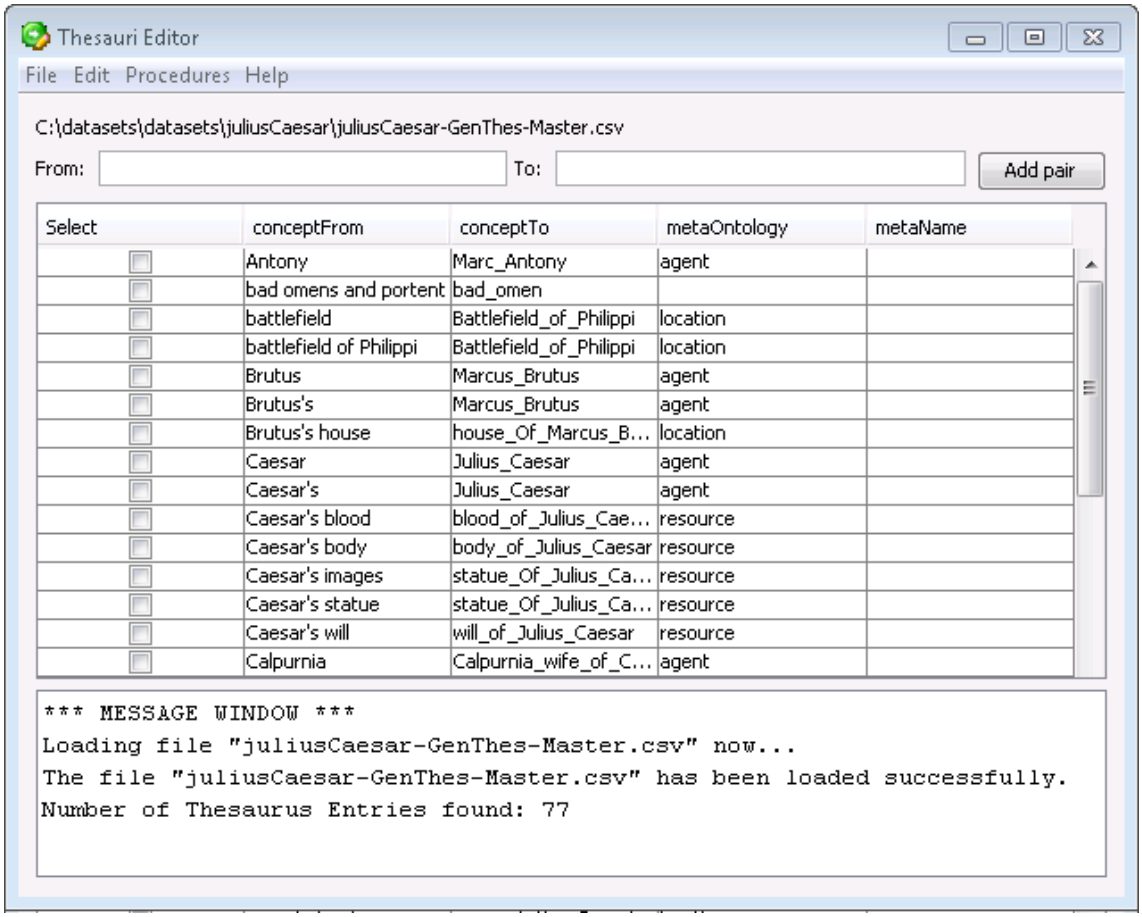
Description

The Thesauri Editor can load and modify existing thesaurus files. Pairs of concepts can be added or subtracted. It can be compared to another thesaurus. Finally it can be saved under a new name.

Under the menus is displayed the name of the currently loaded file. It contains the full pathway of the file.

Below that are the **From:** and **To:** textboxes with the [**Add Pair**] button. This these tools you can add rows to the current file.

The main display conatins five columns. **Select** is used to tell AutoMap which files to run Edit and Procedures on. **conceptFrom** contains the text as it appears in the original file. **conceptTo** is the concept you want to change it to. **metaOntology** contains the class of node. See **Content > Ontology** for more information. **metaName** for future use.



GUI

If you find a pair that does not exist in your thesaurus it can be added by placing the raw text in the To: textbox and the key_concept in the From: textbox. Then click the **Add pair** button to add it to the list.

Sorting

To sort the list click on any of the headers. AutoMap will sort the entire list by the clicked header in an **ascending order**. Clicking that same header again will sort the list in a **descending order**. Clicking a different header will once again sort in an **ascending order**.

NOTE : The small triangle to the right of the header will tell you which header is used for sorting and whether it's in ascending **upward facing arrow** or descending **downward facing arrow** order.

The File Menu

Open File : Select a Thesaurus to load into Editor.

*See **Compare Thesauri Files Lesson** for more information*

Save as... : Saves the Thesaurus.

Save as... : Saves a Thesaurus with a new name and/or to a new directory.

Save message Log : Saves message log form the Thesaurus window.

Convert File to UTF-8 : Attempts to convert an input file into the UTF-8 format.

Exit : Exits the Thesauri Editor and returns to the Main GUI.

The Edit Menu

Compare : Compares a second Thesaurus to the currently loaded Thesaurus.

Add Terms from Concept List : Asks user to select a Concept List which will be added to the currently loaded Thesaurus.

Add Terms from NGram : Asks user to select an NGram List which will be added to the currently loaded Thesaurus.

Add Stemmed Terms : Adds stemmed words to the currently open Thesaurus. The User will be asked whether to use the Porter Stemmer or the K-Stemmer.

Select All : Places a check mark in every box in the **Select** column.

Select None : Removes the check marks from every box in the **Select** column.

Remove Selected : Removes the concepts which contained a check mark in the Select column. The original file remains unaffected.

Identify Possible Misspelling : Highlights in yellow concepts AutoMap may consider misspelled. Hovering over these concepts will give a list of alternatives.

Find : AutoMap asks for term to locate. If there are any matches the background of the found item will be colored blue.

NOTE : In a large thesaurus manually looking through it is usually not an option. Use the **Find** option and type in your search parameters in the textbox. The found item will be displayed with a blue background.

NOTE : Searching for **t*e** would find **the, there, and theatre** (if all three were in your list.

Reset Colors : To end the comparison use **Reset** and all the color bands will be removed.

NOTE : The colors are cleared but any extra cells from the compared file remain on screen. To do a new comparision open a new file.

The Procedures Menu

The functions in this pull-down menu do not affect the currently loaded Delete List. They are identical to the functions that can be found in the Main GUI.

Apply Stemming to Thesauri : You are asked to select a stemmer to apply (Porter Stemmer or K-Stem). All newly stemmed words will be added to the Thesaurus on screen. You need to use one of the **Save options** to keep this new list.,/tr>

Merge Generalization Thesauri : Allows you to select two or more Thesauri and combine them into one. AutoMap will then prompt you to save the new Thesaurus with a new name and location.

Sort Thesaurus : Choose a thesaurus to sort. AutoMap sorts the thesaurus by **number of words** (e.g. the more words in a concept then higher in the list it rises).

Check Thesaurus for Missing Entries : Verifies that each entry in a thesaurus contains no blanks before or after the comma. The line(s) containing the errors will be displayed in the message pane.

Check Thesaurus for Duplicate Entries : Will give the user a notice if there are duplicate entries in a thesaurus.

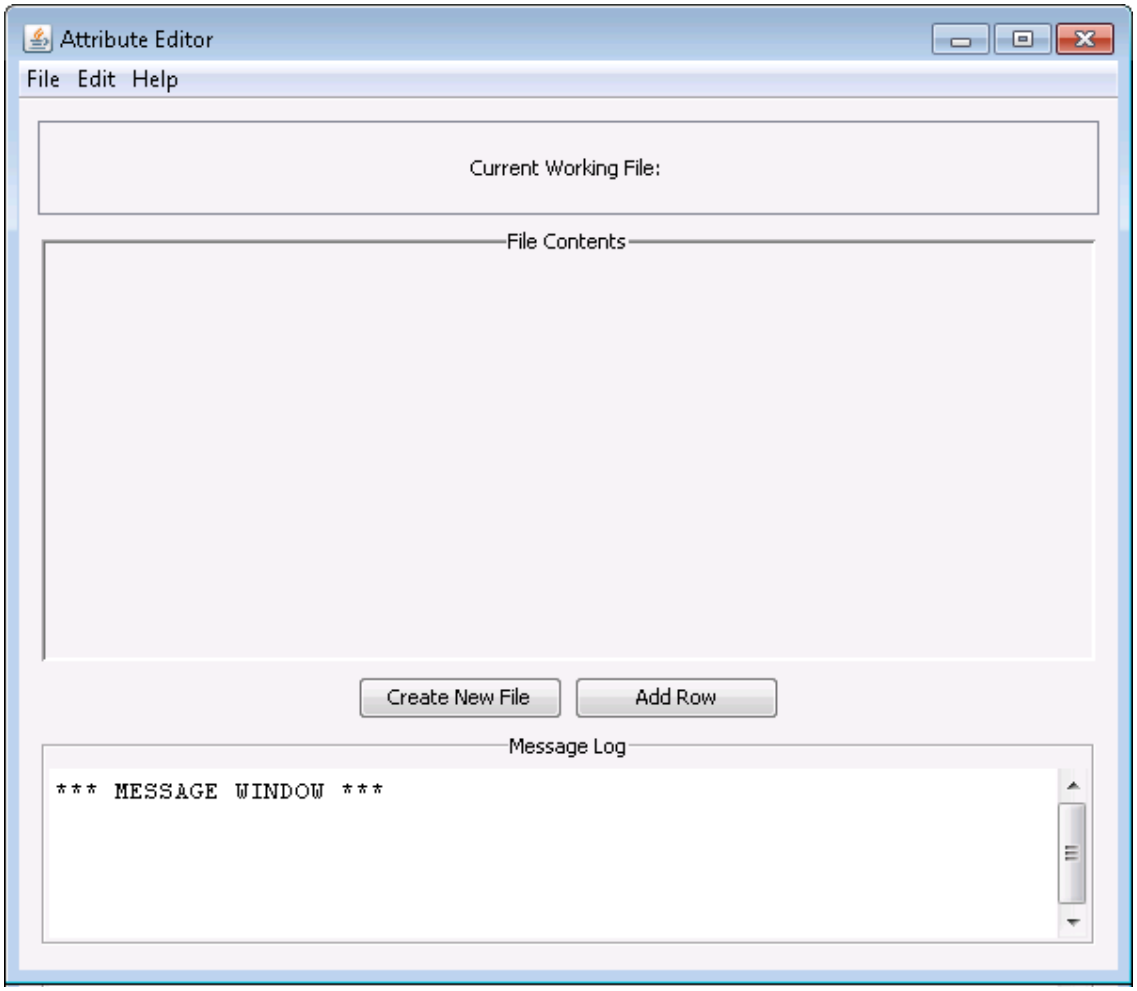
Check Thesaurus for Circular Logic: Will find each instance of Circular Login in a thesaurus and report the line(2) with the problems. Then it will report the total number of instances found.

Check Thesaurus for Conflicting Entries:

19 APR 11

Attribute Editor

The **Attribute Editor** allows you to edit your support files which are in the **.csv** format. (**i.e. Thesauri - Standard and Master format**). New files can also be created from the editor which allows you to control the number and names of the headers.



File Menu

Open File : Opens up a **.csv** file for editing.

Create New File : Creates a new file. Allows you to give your own name column headers. There is no limit to the amount of columns you can create.

NOTE : When creating a Master Thesauri file AutoMap will only recognize columns used by a Master Thesauri.

Save File : If a file was previously opened AutoMap will write a new clumn to the same location. If a new file was created AutoMap will ask for a location to save the file..

Save As : For this function AutoMap **always** asks for a location to save the file.

Save Message Log Window : Saves all activity from the Message Log window.

Exit : Exits the Attribute Editor.

Edit Menu

Compare Files : Asks you to select a file to compare against the currently loaded Attribute file.

Add Terms from Concept List : .

Add Terms from NGram : .

Add Stemmed Terms : .

Select All : Places a check mark in the [**Selected**] column next to every item.

Select None : Removes any check marks in the [**Selected**] columns from all items.

Invert Selection : Places a chek mark in the [**Selected**] column for all unselected items and removes the check mark in the [**Selected**] column from all selected items.

Remove Selected : Deletes all rows with a check mark in the [**Selected**] column.

Find : Highlights all items found which match the search parameter.

NOTE : Will only find exact matches. **Caesar's blood** and **Caesar's body** are not a match and will not be highlighted.

Identify Possible Mispellings : Highlights in orange any items that AutoMap deems might be misspelled.

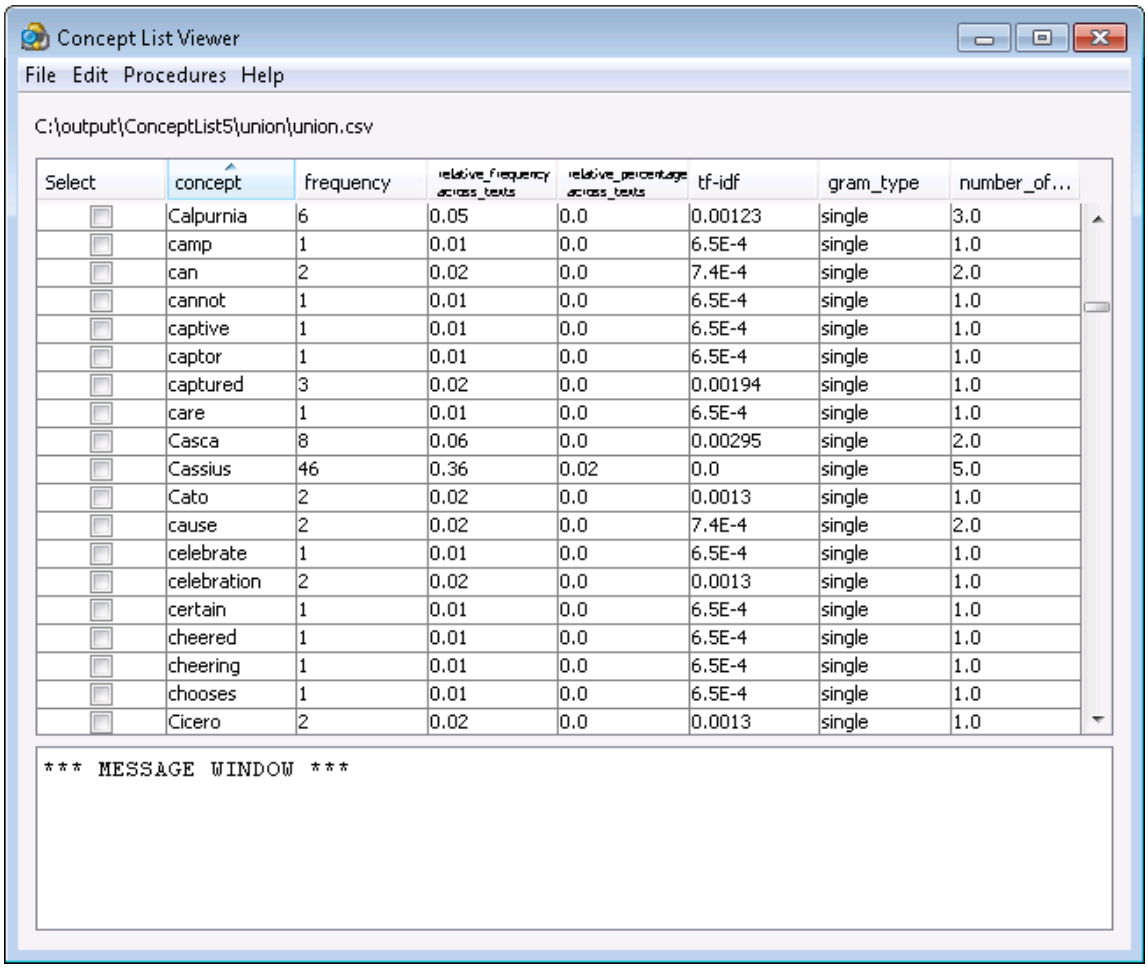
Reset Colors : Removes all highlighting from all items.

19 APR 11

Concept List Viewer

Description

The **Concept List Viewer** is used to view and edit concept lists created from AutoMap. With the viewer you can sort the list by any of the headers. With the **Selected** column you can create a **Delete List**.



Select	concept	frequency	relative_frequency across_texts	relative_percentage across_texts	tf-idf	gram_type	number_of...
	Calpurnia	6	0.05	0.0	0.00123	single	3.0
	camp	1	0.01	0.0	6.5E-4	single	1.0
	can	2	0.02	0.0	7.4E-4	single	2.0
	cannot	1	0.01	0.0	6.5E-4	single	1.0
	captive	1	0.01	0.0	6.5E-4	single	1.0
	captor	1	0.01	0.0	6.5E-4	single	1.0
	captured	3	0.02	0.0	0.00194	single	1.0
	care	1	0.01	0.0	6.5E-4	single	1.0
	Casca	8	0.06	0.0	0.00295	single	2.0
	Cassius	46	0.36	0.02	0.0	single	5.0
	Cato	2	0.02	0.0	0.0013	single	1.0
	cause	2	0.02	0.0	7.4E-4	single	2.0
	celebrate	1	0.01	0.0	6.5E-4	single	1.0
	celebration	2	0.02	0.0	0.0013	single	1.0
	certain	1	0.01	0.0	6.5E-4	single	1.0
	cheered	1	0.01	0.0	6.5E-4	single	1.0
	cheering	1	0.01	0.0	6.5E-4	single	1.0
	chooses	1	0.01	0.0	6.5E-4	single	1.0
	Cicero	2	0.02	0.0	0.0013	single	1.0

Columns

Select : Selected items are the ones AutoMap performas any processing functions on.

concept : Each individual concept is contained on a separate row.

frequency : The amount of occurances found for that concept.

relative_frequency_across_text :

relative_percentage_across_text :

tf-idf :

gram_type :

number_of_texts : The number of texts in which a particular concept is found.

GUI

Sorting

To sort the list click on any of the headers. AutoMap will sort the entire list by the clicked header in an **ascending order**. Clicking that same header again will sort the list in a **descending order**. Clicking a different header will once again sort in an **ascending order**.

NOTE : The small triangle to the right of the header will tell you which header is used for sorting and whether it's in ascending **upward facing arrow** or descending **downward facing arrow** order.

Pull-Down Menus

The File Menu

Open File : Select a Concept List to load into the Viewer.

*See **Compare Concept Lists** lesson for more information.*

Save Message Log : Saves the message log in the Concept List window.

Save as Delete List : Saves check items as a new Delete List.

Exit : Exits the Concept List Viewer and returns to the Main GUI.

The Edit Menu

Compare File : Compares a second Concept List to the currently loaded Concept List.

Properties : Display the **Total Concepts** and the **Unique Concepts** in the loaded file.

Select All : Places a check mark in every box in the **Select** column.

Select None : Removes the check marks from every box in the **Select** column.

Select Minimum Threshold : Selects all concepts with frequencies equal to or greater than the Minimum Threshold.

Select Maximum Threshold : Selects all concepts with frequencies equal to or less than the Maximum Threshold.

Find : AutoMap asks for term to locate. If there are any matches the background of the found item will be colored blue.

NOTE : Searching for **t*e** would find **the, there, and theatre** (if all three were in your list.

Reset Colors : To end the comparison use **Reset** and all the color bands will be removed.

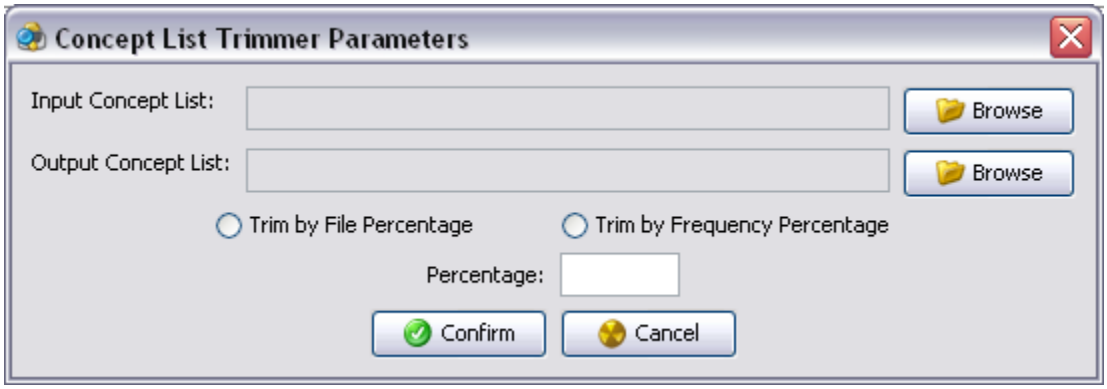
NOTE : The colors are cleared but any extra cells from the compared file remain on screen. To do a new comparision open a new file.

Procedures

Concept List Trimmer : Removes a percentage of the list by one of two methods.

Trim by File Percentage : Trims the concept list by removing lowest frequency items based on percentage of file. Enter 10 and the lowest 10% will be removed.

Trim by Frequency Percentage : Trims the concept list by removing lowest frequency items by based on their frequency as a percentage of highest freqency item. Enter 10 and if the highest frequency item is 100 then every concept that occurs 10 or fewer times will be removed.



11 AUG 11

Table Viewer

Description

The Table Viewer is used to view any **.csv** file. If the file contains headers they will be displayed at the top

Generalization Thesauri - Standard Format

Antony	Marc_Antony
bad omens and portent	bad_omen
battlefield	Battlefield_Of_Philippi
battlefield of Philippi	Battlefield_Of_Philippi
Brutus	Marcus_Brutus
Brutus's	Marcus_Brutus

Generalization Thesauri - Master Format

conceptFrom	conceptTo	metaOntology	metaName
Antony	Marc_Antony	agent	
bad omens and portent	bad_omen		
battlefield	Battlefield_of_Philippi	location	
battlefield of Philippi	Battlefield_of_Philippi	location	
Brutus	Marcus_Brutus	agent	

Agent by Agent Square Matrix

A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16
A1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1
A2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A3	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
A4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

GUI

Sorting

To sort the list click on any of the headers. AutoMap will sort the entire list by the clicked header in an **ascending order**. Clicking that same header again will sort the list in a **descending order**. Clicking a different header will once again sort in an **ascending order**.

NOTE : The small triangle to the right of the header will tell you which header is used for sorting and whether it's in ascending **upward facing arrow** or descending **downward facing arrow** order.

Pull-Down Menus

File Menu

Open File : Navigate to a .csv file to view. If it's a compatible file he information will be displayed in the viewer.

Save Message Log Window : Saves the Message Log Window to the directory of your choice.

Compare File : After selecting your first table you can use this function compare another .csv file. This compare function works slightly different from other compare functions. Instead of examining an individual column to compare it does a one-to-one compare in list order. (e.g. item 1 of file 1 is compared to item 1 of file 2, and so on down the lists).

As in other compare functions a white background means the cell values are identical, a green background means the compare file is a new value, a red background means the compared cell doesn't exist in the loaded file, and a yellow background means the values are different.

 **Exit :** Exits the Table Viewer and returns to the Main GUI.

Edit Menu

 **Compare File :** Compares a second Table to the currently loaded Table.

 **Find :** Highlights in the table the searched for word.

NOTE : Searching for **t*e** would find **the, there, and theatre** (if all three were in your list.

 **Reset Colors :** Resets all colors to black text on white backgrounds.

NOTE : The colors are cleared but any extra cells from the compared file remain on screen. To do a new comparision open a new file.

18 APR 11

XML Viewer

Description

The **DyNetML Network Viewer** allows you to view a DyNetML files properties and relationships. From the pull-down menu select ***Tools => DyNetML Network Viewer***. From the viewer's pull-down menu select ***File => Open File***. Navigate to the xml file to view and click

NOTE : This viewer will open any XML file. It will ignore attempts to open other types of files.

The DyNetML viewer can examine both your semantic network files and your DyNetML files. Each file will display it's structure and the individual properties of the nodes and networks.

GUI

Each section will contain either a **+** or **-** button which will expand or contract that section.

Sorting

To sort the list click on any of the headers. AutoMap will sort the entire list by the clicked header in an **ascending order**. Clicking that same header again will sort the list in a **descending order**. Clicking a different header will once again sort in an **ascending order**.

NOTE : The small triangle to the right of the header will tell you which header is used for sorting and whether it's in ascending **upward facing arrow** or descending **downward facing arrow** order.

Pull-Down Menus

File Menu

Open File : Opens either Semantic or MetaNetwork files and display the file structure.

Save As : You can save the current network to a new directory under a new name.

Exit : Exits the DyNetML Viewer and returns to the Main GUI.

View Menu

Expand : Expands out the entire network.

Collapse : Collapses the entire network.

Procedures Menu

Add Attribute:

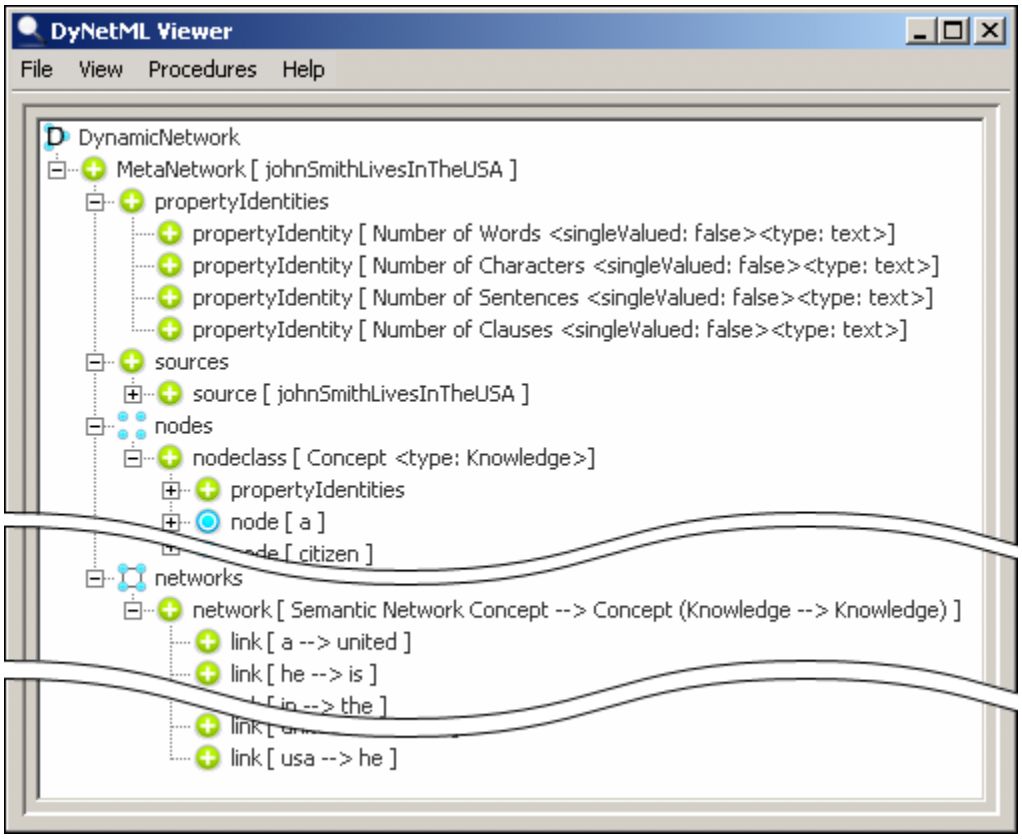
 **Add Attributes:**

 **Relocate Source Location:**

 **Add Icon Reference to DyNetML:**

Network Displays

Displaying a Semantic Network



When viewing a Semantic Network the viewer will display four main areas:

propertyIdentities
Information about the source file, number of words, characters, sentences, and clauses.

sources
The source files in the semantic network

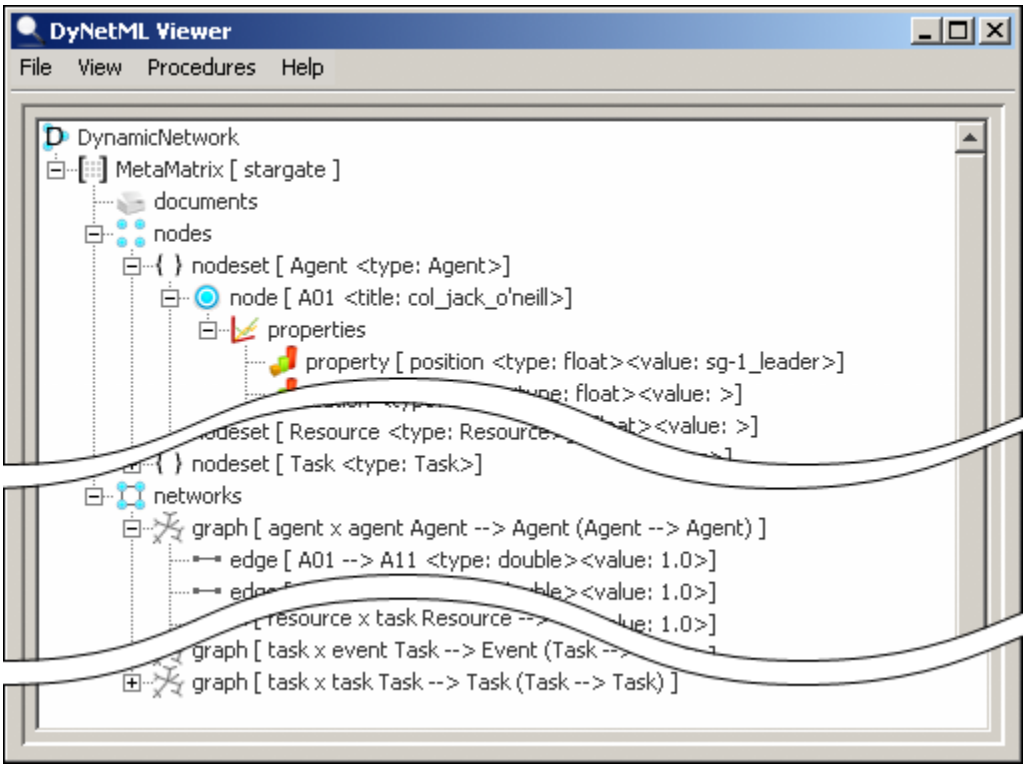
nodes

The nodeclasses in the semantic network and information regarding each nodeclass and node.

networks

Information on each network and the links contained in each network.

Displaying a NetaNetwork



When viewing a Meta-Network (Carley, 2002) the viewer will display two main areas: **nodes and networks**.

nodes

The nodeclasses and the nodes each contains and the properties of each node.

networks

The graphs which make up each network and all the links contained in each network.

29 OCT 09

Description

A viewer that can be used to view text files which have been tagged with either **Parts-of-Speech** or **MetaNetwork** tags.

Parts of Speech Tagged File

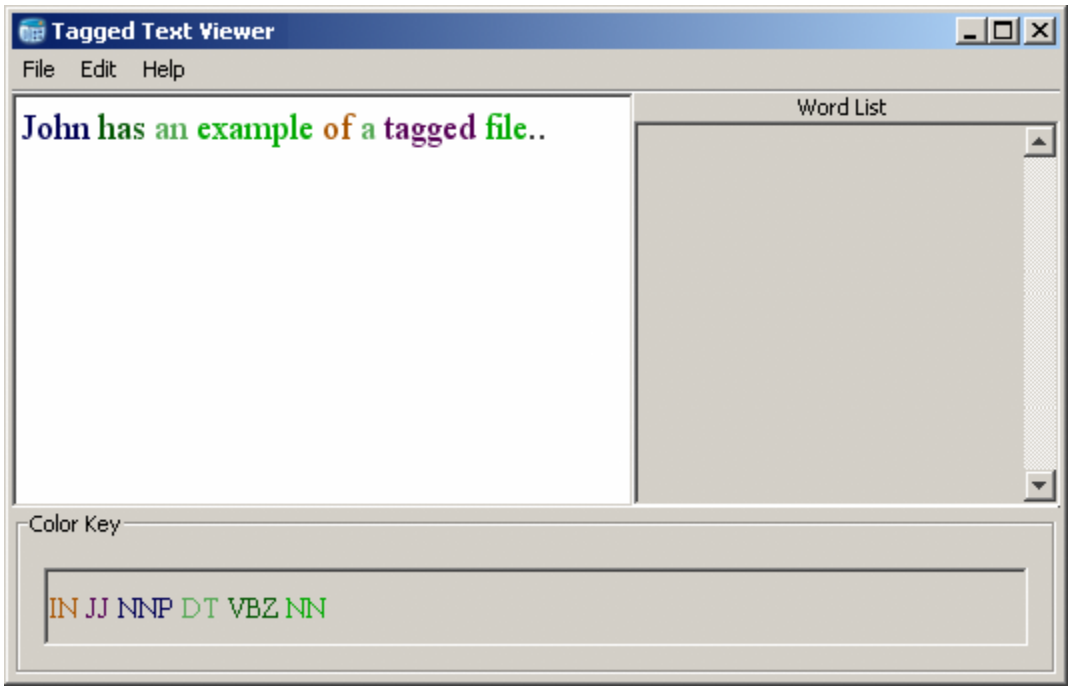
A Parts of Speech tagged file contains tags defining the part of speech of each concept. This is done from the main GUI ***Generate => Parts of Speech Tagging***. The file created can be either in the **.txt or .csv** format. For use in the Tagged Text Viewer you need to save your file in the .txt format.

aTaggedFile.txt

```
John has an example of a tagged file.
```

POS Tags

```
John/NNP has/VBZ an/DT example/NN of/IN a/DT
tagged/JJ file/NN ./.
```



MetaNetwork Tagged File

A MetaNetwork tagged file is generated from the main GUI menu ***Generate => MetaNetwork => MetaNetwork Text Tagging***

. First you will be asked to select a location to save your file. Then you will asked to navigate to a MetaNetwork thesaurus to use.

aTaggedFile.txt

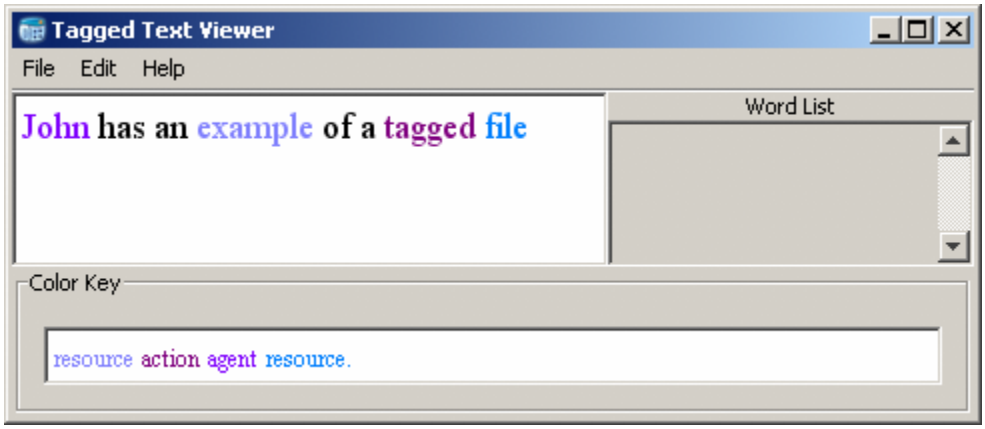
```
John has an example of a tagged file.
```

MetaNetworkThesaurus.csv

```
John,agent
example,resource
tagged,action
file,resource
```

MetaNetwork Tags

```
John/agent has an example/resource of a
tagged/action file/resource.
```



GUI

Word List

A list of words selected from the text is displayed in this pane. **Clicking** any of the words in the display window will place the word in the **Word List panel**.

Color Key

The color coding of the concepts in the display window match the colors of the definitions in the Color Key at the bottom of the window. For a complete list of the Parts of Speech see **Content => Parts of Speech**.

The Color Key can also be used to highlight various parts of speech in your text. By clicking on the parts of speech in the Color Key the coresponding taged concepts will be highlighted in the text window.

No Selections

Select nouns (NN)

Select proper nouns (NNP)

Pull-Down Menus

File Menu

Open File : Loads a text file into the viewer.

Save as HTML : Saves the current file in the HTML format. This file can be used for demonstration purposes or for purposes of further analysis.

Save Checked Words to File : Saves all checked words to a list.

Add Checked Words to Delete List : Places all checked words in a text file as a list with one word per line.

Add Checked Words to MetaNetwork Thesaurus : AutoMap asks you to classify the checked words and create a comma-delimited file. All the checked items will receive the same classification.

Exit : Exits the Tagged Text Viewer and returns to the Main GUI.

Edit Menu

Remove Checked Words : Removes from the Word List all words that have been checked.

Remove All Words : Removes all words from the Word List regardless of whether or not they are checked.

Identify Possible Misspellings : Italicizes all words it deems as possible misspellings.

Find Word : Makes any instance of the found word bolds. This function can be repeated multiple times and previous found words will remain in bold. Use **Reset Colors** to clear.

NOTE : Searching for **t*e** would find **the, there, and theatre** (if all three were in your list.

Reduce Deleted Words : Makes any instance of a deleted word reduced in size.

Regular display

synopsis: xxx tok_ra plan xxx kill xxx xxx system_lords. xxx plan xxx xxx infiltrate xxx summit xxx poison xxx system_lords

Reduced display

synopsis: xxx tok_ra plan xxx kill xxx xxx system_lords. xxx plan xxx xxx infiltrate xxx summit xxx poison xxx system_lords

Show Delete List Impact : Asks for a Delete List to apply and will display, by strike-through, how that Delete List would affect the file.

Show Generalization Thesaurus Impact : Asks for a Generalization Thesaurus and will display, by underlining adjacent concepts, how that thesaurus would affect the file.

Show MetaNetwork Thesaurus Impact : Asks for a MetaNetwork Thesaurus and will display, by color coding found concepts.

Set Font Size : Changes the font size using HTML sizes, not point sizes.

Set Font Style : Allows you to change the display to any font on your computer.

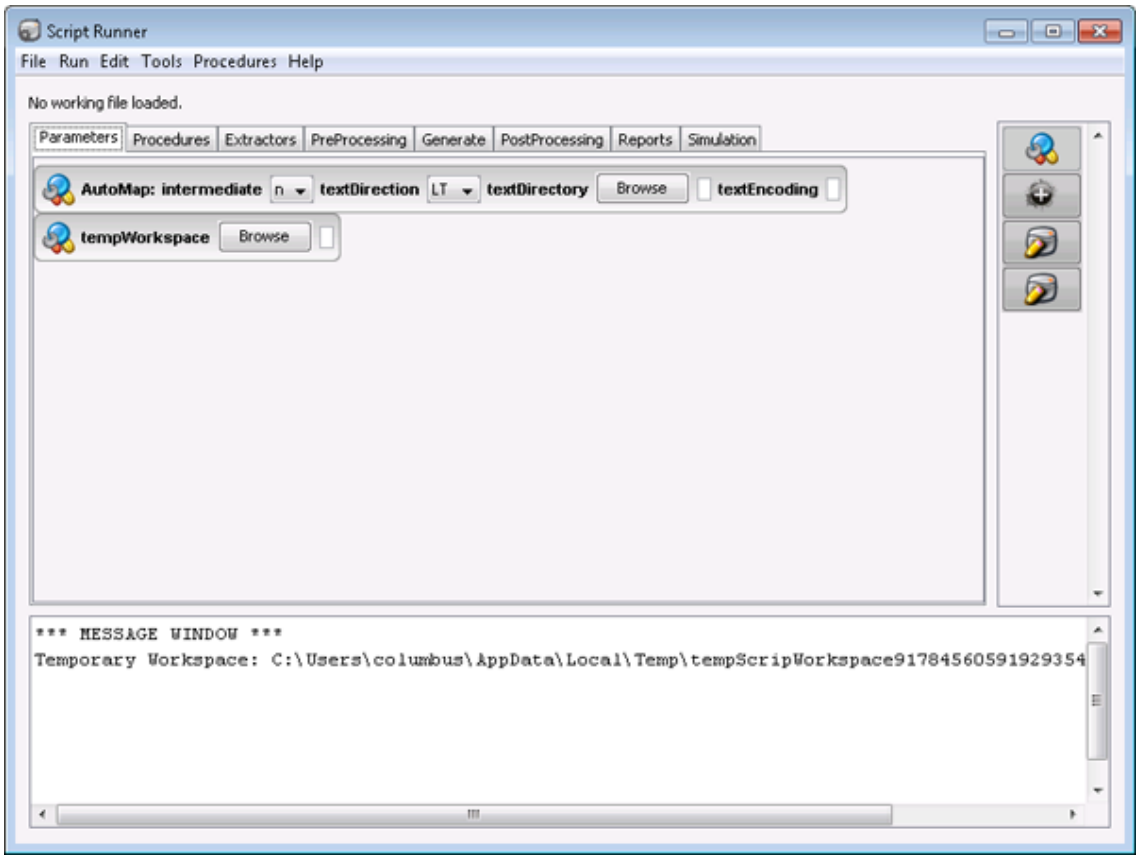
Reset : Resets all colors and font styles in the display to their defaults.

19 APR 11

Script Runner

Script Runner is explicitly used to process large sets of data from parameters tested from running a limited set of data in the GUI. After creating and modifying a set of functions in the GUI you can use those parameters to create you .aos file in order to process large sets.

And after a script is created and loaded again, many of the functions can be altered to obtain a different set of results ***(e.g. change the Delete List run on a set of files)***.



GUI

The GUI consists of four parts. 1) The Menus; 2) The Tabs; 3) The Quick Launch buttons; and 4) The Message Window.

Menus

File Menu

Load Script File : Loads a script file either created in an external program or created previously in Script Runner.

New Script File : Create a New Script file from scratch

Save : Saves currently loaded script file

Save As... : Saves curently loaded script file that can be renamed as new file.

Save Message Window Log : As AutoMap is running your script it will display details on the actions it has performed. You can save these messages to a text file.

Run

Run This Script File : Runs the script currently loaded in the viewer pane.

Run This Script File as SuperScript : Runs the script currently loaded in the viewer pane under multiple processors

Script 2 BPEL : Converts a file from ScriptRunner into a format usable by the SORASCS server.

Edit

Suggest Variables :

Suggest Temporary Directory :

Preprocess Script File :

Script 2 Package :

Tools

In addition to running scripts the Script Runner tool can call up other viewers. These can be used to verify the state of your files before or after running a script without leaving the viewer.

Delete List Editor : Calls the external Editor to work with a Delete List.

See **Tools => Delete List Editor** for more information.

Thesaurus Editor : Calls the external editor to work with a Thesaurus file.
See **Tools => Thesaurus Editor** for more information.

Concept List Viewer : Calls the external viewer to review a Concept List
See **Tools => Concept List Viewer** for more information.

Table Viewer : Allows the user to view table files other than Concept Lists and DyNetML files.
See **Tools => Table Viewer** for more information.

XML Network Viewer : Allows the user to view DyNetML and other XML.
See **Tools => XML Viewer** for more information.

Tagged Text Viewer :
See **Tools => XML Viewer** for more information.

Script Config :

Add Plugin :

Procedures

Run a Script File : Navigate to the .config file you want to run. This can be a script you created in a text editor or a script created from AutoMap's main GUI pull-down menu **File => Save Script File** which will create a script of all current preprocessing steps.

Run a Script File as SuperScript : Allows user to run a script under multiple processors. User inputs the number of processors to use and AutoMap splits the input files into that many batches.

Script Runner Tabs

The tabs at the top of the window are performed from left to right and all functions within a specific window are performed from top to bottoms. They include:

Parameters : Maintains information on the workspace and other information about the files being processed.

Procedures : Functions to prepare data files and support files which includes merging Delete LIsts and Thesauri files.

Extractors : Used to get information from sources other than standard text files which includes FacebOok, Blogger, Twitter, and RSS feeds.

PreProcessing : Includes all the Preprocessing functions found in the GUI which includes Delete List, Thesauri, and various removal functions.

Generate : After all PreProcessing is finished these functions **generate** some type of output which includes Semanatic List, Meta-Networks, and other lists of concepts.

PostProcessing : Works on generated files to further process them which includes attributes, beliefs, and unions.

ReportsContains the reports useful after all processing is complete on text files.

Simulation

Quick Launch Buttons

The set of buttons will change when a different tab is selected. The buttons will be functions needed for each different function.

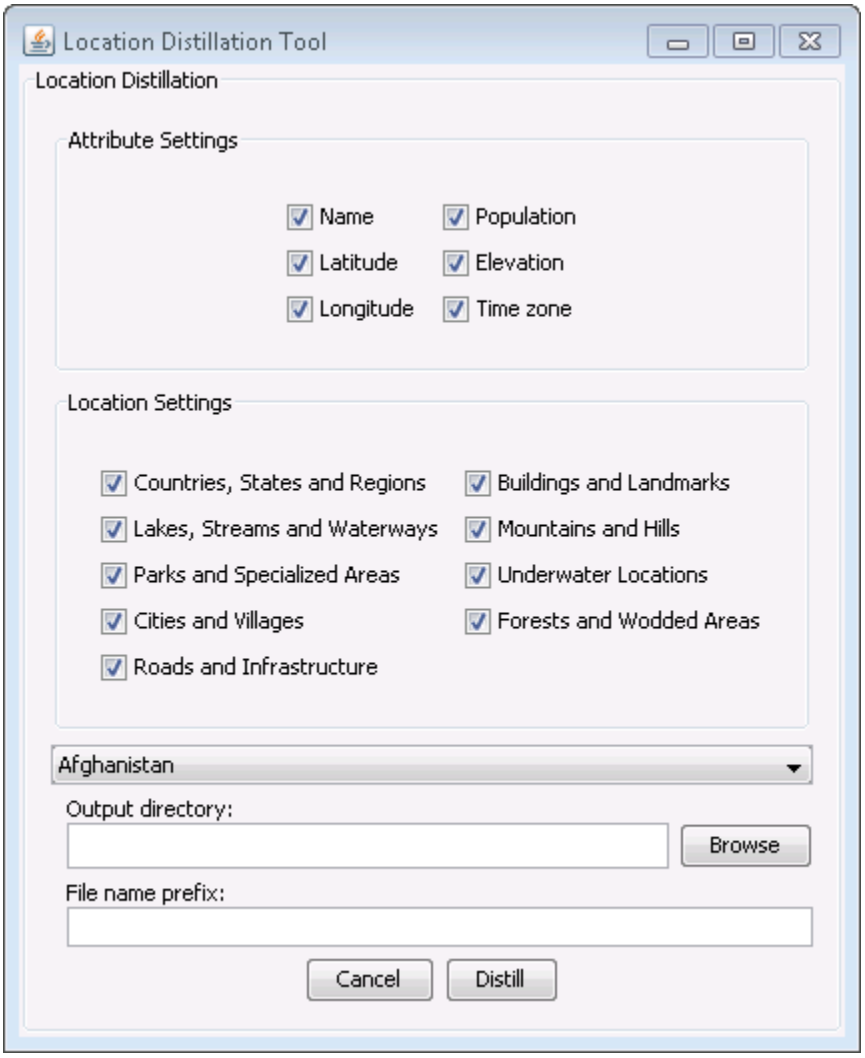
Message Window

Keeps track of all the user's actions and is also editable. In addition the message window can be saved.

Location Distillation

A review of the dialog box will show you what information AutoMap can detect from your files. It uses the **allCountires.txt** file as its source.

NOTE : If AutoMap does not find this file you will be directed as to how to download it.



The Location Distillation will pull out every reference in the file for every category check marked. Remove the check marks from the settings which you will not need in order to reduce the size of the final file.

NOTE : If you need information about separate countries you need to run the process once for each location. Then you can merge these individual thesauri together.

This creates a file that can be used as a base thesauri.

27 MAY 11

Compare Color Chart

During a **Compare File** function AutoMap will color the background of various concepts to visually mark the state of a concept. The following chart explains what the colors mean.

Color	Description
Red	Concepts to be deleted after comparison
Green	Concepts to be added after comparison
Yellow	Concepts to be modified after comparison
Orange	Possible misspelled terms
Cyan	Concepts found during a dource
Pink	Terms added from stemming
Grey	Duplicate entries

Only the colors necessary for any particular tool will appear in the comparison tables. For instance if there is no stemming option then no magenta cells will ever appear.

30 OCT 09

Text Partitioner

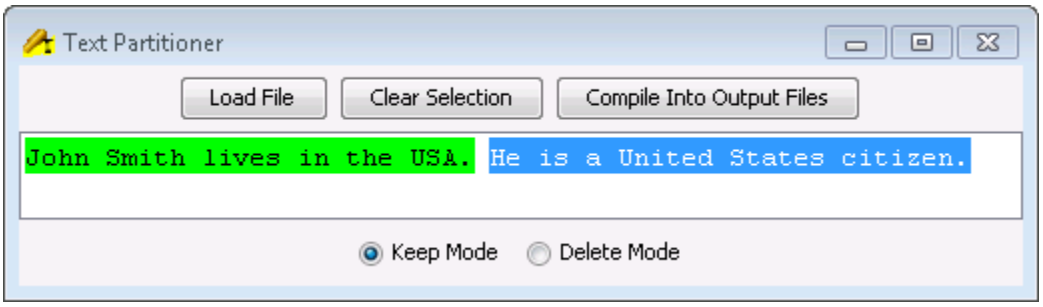
Description

Divides one file into multiple smaller files. Each separate highlighed sections is output as a new file.

Procedure

Load File : Click to select a file to partition. You can now select individual sections of the text which will alternate in green and blue highlighting.

> **NOTE :** These color do nothing special. They are only used to assist you in seeing where your divisions are place.



Clear Selection : If you find you've divided you file incorrectly use the [**Clear Selection**] button to remove all highlighting.

> **NOTE :** Clicking this button removes **ALL** selections.

Compile Into Output File : After clicking navigate to a directory to save your files. One file will be written for each highlighted section.

Buttons

Keep Mode : Highlights selected text in alternating blue and green. The colors mean nothing and are only used to help you see your selections.

Delete Mode : Highlights text in red to alert you to the fact you've designated that text not to be included.

> **NOTE :** Once text has been designated as being **kept** it can not be designated to be **deleted**.

Script

Description

All of AutoMap's functions are readily found in the Script file. A few items are necessary when using the script.

AM3 Script Notes

AM3 Script Tags

DOS Commands

Things You Need To Know

1. Knowledge of the Command Run Window.
2. Understanding of XML formatting.
3. DOS Commands

21 AUG 09

AM3Script Notes

Using AutoMap 3 Script

The AutoMap 3 script is a command line utility that processes a large number of files using a set of processing instructions provided in the configuration file. Following is a simple explanation of how to construct a configuration file.

Once the configuration file has been created, the Automap 3 Script is ready to use. The following is a brief on running the script.

1. Configure the **AutoMap 3 .aos file** as necessary. (Tag explanations in next section). Be sure to include pathways to input and output directories and the name of the config file to use.

```
<Settings>
```

```
<AutoMap
    textDirectory="C:\My
    Documents\dave\project\input"
    tempWorkspace="C:\My
    Documents\dave\project\output"
    textEncoding="unicode"/>
</Settings>
```

2. Navigate to where AutoMap is installed.
3. At the prompt type: **am3script newProject.aos** (where newProject.aos is the config file you built).
4. AutoMap 3 will execute the script using the .aos file specified.

For Advanced Users

It is possible to set the your PATH environmental variable to include the location of the install directory so that AM3Script can be used in any directory from the command line. Please note this is not recommended for users that have no experience modifying the PATH environmental variable.

Placement of Files

It is suggested the user create sub-directories for input files and output files in within an overall directory. This assists in finding the correct files later and prevents AutoMap from overwriting previous files. The **input** directory is empty except for your text files. The **output** will contain the output from AutoMap. The **support** directory will contain your Delete Lists, Thesauri, and any other files necessary during the run.

```
C:\My Documents\dave\project\input
C:\My Documents\dave\project\output
C:\My Documents\dave\project\support
```

NOTE : It's important when typing in pathways that they are correct or AutoMap will fail to run.

Script name

The script.aos file can be named whatever you like but we do recommend keeping the `.aos` suffix. This way if you can do multiple runs to the files in a concise order: `step1.aos, step2.aos, step3.aos...`.

Pathways

Pathways used in attributes are always relative to the location of AM3Script, (e.g. `/some_files` uses a directory `some_files` below the directory AM3Script is located in. A full pathway always begins with the drive name e.g. `C:/` and follows the pathway down to the files.

NOTE : Both relative and absolute paths can be used for the configuration path. Relative traces a path from the location the config to the file it needs **(e.g. ..\..\anotherDirectory/aFile)**. Absolute traces a pathway from the root directory to the file it needs **(C:\\{pathway}\aFile)**.

If given a non-existent pathway you will receive an error message during the run.

Tag Syntax in AM3Script

There are two styles of tags in the AM3Script script. The first one uses a set of two tags. The first tag starts a section and the second tag ends the section. The second tag will contain the exact same word as the first but will have, in addition, a "/" appended after the word and before the ending bracket. This designates it as an ending tag. All the parameters/attributes pertaining to this tag will be set-up between these two tags. e.g. **<aTag></aTag>**.

The second style is the self-ending tag as it contains a "/" within the tag. Any attributes used with this tag are contained within the tag e.g. **<aTag attribute="attributeName"/>**.

Output Directory syntax (TempWorkspace)

Output directories created in functions under the <PreProcessing> tag will all be suffixed with a number designating the order they were performed in. If a function is performed twice, each will have a separate suffix i.e. Generalization_3 and Generalization_5 denotes a Generalization Thesauri was applied to the text in the 3rd and 5th steps. Using thesauriLocation different thesauri could be used in each instance. For all other functions outside PreProcessing there is no suffix attached.

NOTE : The output directories specified above are in a temporary workspace and the content will be deleted if the AM3Script uses this directory again in processing. It is recommended that the directory specified in the temp workspace be an empty directory. Also, for output that user wishes to keep from processing it is recommend to use the outputDirectory tag within the individual processing step.

Example

```
<AddAttributes3Col attributeFile="C:\My
Documents\dave\project\support\attributeFile"
outputDirectory="C:\My
Documents\dave\project\output" />
```

By using these tags it allows the user to specify where they want the individual processing step output to go. It also makes finding the location of the output files much simpler instead of looking through the contents of the TempWorkspace.

AutoMap 3 System tags

The only line found outside these tags will be the declaration line for xml version and text-encoding information: `<?xml version="1.0" encoding="UTF-8"?>`

NOTE : Any parameter can use inputDirectory and outputDirectory to override the default file location. These pathways will be relative to the location of the AM3Script.

18 AUG 09

AM3Script Tags

NOTE : Note that every tag can have an additional outputDirectory="" element added to permanently save file location. If the script is crashing on you, it may be because you aren't saving some output you've generated (like POS) and Automap wants to access it. Try running again and saving the output.

19 AUG 09

AM3Script Tags-Script

NOTE : Note that every tag can have an additional outputDirectory="" element added to permanently save file location. If the script is crashing on you, it may be because you aren't saving some output you've generated (like POS) and Automap wants to access it. Try running again and saving the output.

<Script>

<Settings></Settings>

<AutoMap>

`textDirectory :` Pathway to the directory containing your text files to process.

`tempWorkspace :` Directory for storing files while processing. Files in this directory are **NOT** automatically deleted.

`textEncoding :` Includes autoDetect.

`intermediate :` Set **intermediate="y"** to tell AutoMap that processing has been performed on your text. **intermediate="n"** tells AutoMap you are working with raw text.

`textDirection :` **LT | RT | LB | RB** chooses the started point for the text. They stand for **Left/Top - Right/Top - Left/Bottom - Right Bottom**

3 MAY 11

AM3Script Tags-Extractors

NOTE : Note that every tag can have an additional outputDirectory="" element added to permanently save file location. If the script is crashing on you, it may be because you aren't saving some output you've generated (like POS) and

Automap wants to access it. Try running again and saving the output.

<Extractors><Extractors />

<YahooExtractor />

`search :`

`printnumresults :` **y | n**

`firstindex :`

`results :`

`usetitle :` Set **usetitle="y"** to include the title of the web page. **usetitle="n"** excludes the title.

`region :` The country to search.

`type :` **all | phrase | any**

`language :` The language of the web sites.

`site :`

`format :` **any | html | msword | pdf | ppt | rss | txt | xls**

`similarok :` **y | n**

<ExcelConverter/>

<PdfConverter/>

<PowerPointConverter/>

<WordDocConverter/>

<WebScraper />

`url :` The address for the web page to extract. Requires the complete protocol.

AM3Script Tags-PreProcessing

NOTE : Note that every tag can have an additional outputDirectory="" element added to permanently save file location. If the script is crashing on you, it may be because you aren't saving some output you've generated (like POS) and Automap wants to access it. Try running again and saving the output.

<PreProcessing></PreProcessing>

<DedupeText />

<DeleteList />

`adjacency :` Set **adjacency="d"**, for direct which completely removes words. Remaining concepts now become "adjacent" to each other. Set **adjacency="r"** for rhetorical which removes the concepts but inserts a spacer (**XXX**) within the text to maintain the original distance between concepts.

`deleteListLocation :` Location to save final Delete List

<FilterDirectory />

`filter :`

<FormatCase />

`changeCase :` Changes the output text to either lowercase **changeCase="l"** or uppercase **changeCase="u"**.

<Generalization />

`thesauriLocation :` Location of final thesauri file

`useThesauriContentOnly :` Set **useThesauriContentOnly="n"** and AutoMap replaces concepts in the Generalization Thesauri but leaves all other concepts intact. Set **useThesauriContentOnly="y"** and AutoMap replaces concepts but removes all other concepts from output file.

<PdfConverter />

<PronounResolution />

<RemoveExtraWhiteSpace />

Find instances of multiple spaces and replaces them, in total, with a single space.

<RemoveNumbers />

This parameter accepts either **whiteOut="y"** or **whiteOut= "n"**. A **y** replaces numbers with spaces

EXAMPLE : whiteOut="y" replaces numbers with spaces (C3PO => C PO). **whiteOut="n"** removes the numbers entirely and closes up the remaining text (C3PO => CPO).

<RemovePunctuation />

`whiteOut :` **whiteOut="y"** replaces punctuation with spaces. **whiteOut="n"** removes the punctuation entirely and closes up the remaining text. The list of punctuation removed is: **.,:;' "()!?-**.

<RemoveSpecialCharacters />

<RemoveSymbols />

`whiteOut :` **whiteOut="y"** replaces punctuation with spaces. **whiteOut="n"** removes the punctuation entirely and closes up the remaining text. The list of symbols that are removed: **_+={}[]\|/<>**.

<RemoveUserSymbols />

`symbols :` Similar to **RemoveSymbols** except it allows you to choose the symbols to remove. Place the list of symbols to remove in the **symbols** parameter leaving no spaces in-between the symbols.

<Stemming />

Stemming removes suffixes from words. This assists in counting similar concepts in the singular and plural forms.

i.e. plane and planes would normally be considered two terms. After stemming planes becomes plane and the two concepts are counted together.

`type` : **type="k"** uses the KSTEM or Krovetz stemmer. **type="p"** uses the Porter Stemming.

`porterLanguage` : If **type** is set to Porter then you can set the language to any of the following: **Danish, Dutch, English, Finnish, French, German, Italian, Norwegian, Portuguese, Russian, Spanish, and Swedish**

`kStemCapitalization` : **kStemCapitalization="y"** tells AutoMap to stem capitalized words while **kStemCapitalization="n"** ignores capitalized words.

NOTE : If you select Porter Stemming then a language **MUST** be choosen or the script will error.

<VibesParser />

<WebScraper />

`url` : You provide a URL address making sure to use the proper protocol (e.g. http://). It will create text files from all files located on the base address.

NOTE : This will convert all files found from the address downwards meaning that a simple looking URL might possibly contain hundreds, or even thousands, of sub files which will be converted.

<WordDocConverter />

3 MAY 11

AM3Script Tags-Processing

NOTE : Note that every tag can have an additional outputDirectory="" element added to permanently save file location. If the script is crashing on you, it may be because you aren't saving some output you've generated (like POS) and

Automap wants to access it. Try running again and saving the output.

<Processing></Processing> or <Generate></Generate>

<Anaphora />

An anaphoric expression is one represented by some kind of deictic, a process whereby words or expressions rely absolutely on context. Sometimes this context needs to be identified. These definitions need to be specified by the user. Used primarily for finding personal pronouns, determining who it refers to, and replacing the pronoun with the name.

<CRFSuggestion />

This option automatically estimates mapping from text words from the highest level of pre-processing to the categories contained in the Meta-Network.

<ConceptList />

Creates a list of concepts for each loaded text file. A Delete List or Generalization Thesauri can be performed before creating these lists to reduce the number of concepts in each file. These output files can be loaded into a spreadsheet and sorted by any of the headers.

<FeatureExtraction />

The Feature Selection creates a list of concepts as a TD*IDF (Term Frequency by Inverse Document Frequency) in descending order. This list can be used to determine the most important concepts in a file. It's used to extraction dates and currency from text files.

<KeyWordInContext />

A list will be created so every concept in a file along with the concepts which both precede it and following it.

<MetaNetText />

<MetaNetwork />

`thesauriLocation :` Applies the Generalization Thesaurus specified in `thesauriLocation` to the text files. Then creates a MetaNetwork using the following four parameters.

`directional :` Can be set to either **directional="U"** for uni-directional or **directional="B"** for bi- directional. Determines whether AutoMap checks in both directions.

`resetNumber :` Set to the number of text units to process before resetting back to 1. Default is **resetNumber="1"**

`textUnit :` Sets the text unit to [**w**]ord, [**c**]lause, [**s**]entence, [**p**]aragraph, or [**a**]ll. The default is **textUNit="s"**.

`windowSize :` Sets the amount of concepts to be considered for replacement. The default value is **windowSize="5"**.

<MetaNetworkList />

`thesauriLocation :` This associates text-level concepts with Meta-Network (Carley, 2002) categories [**agent, resource, knowledge, location, event, group, task, organization, role, action, attributes, when**]. Concepts can be translated into several Meta-Network categories. `thesauriLocation` designates the location of the MetaNetwork (Carley, 2002) Thesauri, if used.

<NGramExtraction />

`createUnion :` Set to **createUnion="y"** to create a union file or **createUnion="n"** to ignore creation of a union.

`ngram :` Default value is **ngram="2"**

<NamedEntityExtraction />

Extracts proper names, numerals, and abbreviations from the texts loaded.

<POSExtraction />

`posType :` You can specify either **posType="ptb"** to tag for each part of speech or **posType="aggregate"**to group many categories together thus using fewer Parts-of-Speech tags.

`saveOutputAs :` The final file is specified as either **saveOutputAs="csv"** or **saveOutputAs="txt"** file.

<PosAttributeFile />

<PositiveThesauri />

Takes every concept in the text and defines it as itself. This can be used as the start in building a Generalization Thesaurus.

<SemanticNetwork />

`directional :` Can be set to either **directional="U"** for uni-directional or **directional="B"** for bi- directional. Determines whether or not AutoMap checks in both directions.

`resetNumber :` Set to the number of text units to process before reseting back to 1. Default is **resetNumber="1"**

`textUnit :` Sets the text unit to]**w**]ord, [**c**]lause, [**s**]entence, [**p**]aragraph, or [**a**]ll. The default is **textUnit="s"**.

`windowSize :` Sets the amount of concepts to be considered for replacement. The default value is **windowSize="5"**.

<SemanticNetworkList />

`directional :` Can be set to either **directional="U"** for uni-directional or **directional="B"** for bi- directional. Determines whether or not AutoMap checks in both directions.

`resetNumber :` Set to the number of text units to process before reseting back to 1. Default is **resetNumber="1"**

`textUnit :` Sets the text unit to **w**ord, **c**lause, **s**entence, **p**aragraph, or **a**ll. The default is **textUnit="s"**.

`windowSize :` Sets the amount of concepts to be considered for replacement. The default value is **windowSize="5"**.

<UnionConceptList/>

Union Concept Lists consider concepts across all texts currently loaded, rather than only the currently selected text file. It reports total frequency, related frequency, and cumulative frequencies of concepts in all text sets. It's helpful in finding frequently occurring concepts over all loaded texts.

NOTE : The number of unique concepts considers each concept only once, whereas the number of total concepts considers repetitions of concepts.

<UnionKeyWordInContext />

3 MAY 11

NOTE : Note that every tag can have an additional outputDirectory="" element added to permanently save file location. If the script is crashing on you, it may be because you aren't saving some output you've generated (like POS) and Automap wants to access it. Try running again and saving the output.

The following tags may occur in any of the main sections:

<print msg="" />

<run path="" args="" />

<Procedures></Procedures>

<ConvertFileEncoding />

`inputFile :` Location of the file you want to convert.

`outputFile :` Location and filename of the newly converted file.

NOTE : The input file remains unchanged.

<MergeDeleteLists />

`deleteListFiles :` The directory containing all the Delete Lists to merge.

`outputDeleteListFile :` The location and filename of the newly merged Delete List.

<MergeThesauri />

`thesauriFiles :` The directory containing all the thesauri to merge.

`outputDeleteListFile :` The location and filename of the newly merged thesauri.

<SortThesaurus />

`thesauriFile :` The location of the thesaurus you want to sort.

`outputThesaurusFile :` The location and filename of the newly sorted thesauri.

<SyntaxParser />

`outputDirectory :` Location to write the newly parsed file.

<ApplyDeleteListToConceptList :

`deleteListLocation :` Location of folder containing Delete Lists

`inputConceptList :` Location of folder containing Concept Lists.

`outputConceptList :` Location of new Concept List

<RemoveNumbersFromConceptList />

`inputConceptList :` Location of folder containing Concept Lists

`outputConceptList :` Location of newly created Concept Lists

<LocationDistillation />

`allCountriesLocation :` Location of the **allCountries.txt** file.

`countryInfoLocation :`

`countryName :` Name of the country to search.

`outputDirectory :` Location to write newly created files

`fileNamePrefix :` Prefix that can be attached to new files.

<MasterThesaurusMerge />

`originalThesaurus :` Location of thesauri to change.

`changeThesaurus :` Location of the Change Thesauri

`outputThesaurus :` Location to write newly changed thesauri

<ConvertDeleteListToMasterThes />

`inputDeleteList :` Location of Delete List to convert

`outputThesaurus :` Location to write newly converted Delete List

<ConvertGenThesToMasterThes />

`inputThesaurus :` Location of Thesauri ro convert

`outputThesaurus :` Location to write newly converted Thesaui

<ConvertMetaThesToMasterThes />

`inputThesaurus` : Location of Meta-Netwotk to convert

`outputThesaurus` : Location to write newly converted Meta-Network.

3 MAY 11

AM3Script Tags-Post-Processing

NOTE : Note that every tag can have an additional outputDirectory="" element added to permanently save file location. If the script is crashing on you, it may be because you aren't saving some output you've generated (like POS) and Automap wants to access it. Try running again and saving the output.

<PostProcessing></PostProcessing>

<AddAlias />

`aliasFile` :

`nodeType` :

<AddAttributes />

`attributeFile` : Additional attributes can be added to the nodes within the generated DyNetML file. **attributeFile** is the location of the attribute file containing a header row with the attribute name.

<AddAttributes3Col />

`attributeFile` : The location of the file containing the attributes but uses **name and value** headers.

<AddTimePeriod />

<BeliefEnhancement />

`beliefFile` :

networkType : **networkType="m" networkType="s"**

<BeliefPropagationReport />

inputFile :

beliefFile :

reportName :

<ClickIt />

networkFile :

outputFile :

location :

<ImmediateImpactReport /.

inputFile :

nodeFile :

reportFile :

<InferredBeliefs />

beliefFile :

<OraReports />

reportType :

reportName :

nodeType :

nodeID :

<PictureIt />

networkFile :

outputFile :

imageDirectory :

preserveExistingImages="y|n" :

<TimeUnion />

unionType="s|m" :

startDate :

endDate :

timeInterval :

<UnionDynetml />

unionType="s|m" : Creates a union of all dynetml in a specified directory. It requires a **unionType** which can be **"s"** for a union of semantic networks or **"m"** for union of metanetworks.

3 MAY 11

DOS Commands

Description

A short description of some DOS commands that can be useful when using the Script.

CD: Change Directory

cd

Goes to the highest level, the root of the drive.

cd..

Goes back one directory. For example, if you are within the C:\Windows\COMMAND> directory, this would take you to C:\Windows>

The CD command also allows you to go back more than one directory when using the dots. For example, typing: cd... with three dots after the cd would take you back two directories.

cd windows

If present, would take you into the Windows directory. Windows can be substituted with any other name.

cd\windows

If present, would first move back to the root of the drive and then go into the Windows directory.

cd windows\system32

If present, would move into the system32 directory located in the Windows directory. If at any time you need to see what directories are available in the directory you're currently in use the dir command.

cd

Typing cd alone will print the working directory. For example, if you're in c:\windows> and you type the cd it will print c:\windows. For those users who are familiar with Unix / Linux this could be thought of as doing the pwd (print working directory) command.

DIR: Directory

Lists all files and directories in the directory that you are currently in.

dir /ad

List only the directories in the current directory. If you need to move into one of the directories listed use the cd command.

dir /s

Lists the files in the directory that you are in and all sub directories after that directory, if you are at root "C:\>" and type

this command this will list to you every file and directory on the C: drive of the computer.

dir /p

If the directory has a lot of files and you cannot read all the files as they scroll by, you can use this command and it will display all files one page at a time.

dir /w

If you don't need the info on the date / time and other information on the files, you can use this command to list just the files and directories going horizontally, taking as little as space needed.

dir /s /w /p

This would list all the files and directories in the current directory and the sub directories after that, in wide format and one page at a time.

dir /on

List the files in alphabetical order by the names of the files.

dir /o-n

List the files in reverse alphabetical order by the names of the files.

dir \ /s |find "i" |more

A nice command to list all directories on the hard drive, one screen page at a time, and see the number of files in each directory and the amount of space each occupies.

dir > myfile.txt

Takes the output of dir and re-routes it to the file myfile.txt instead of outputting it to the screen.

MD: Make Directory

md test

The above example creates the **test** directory in the directory you are currently in.

md c:\test

Create the **test** directory in the c:\ directory.

RMDIR: Remove Directory

rmdir c:\test

Remove the test directory, if empty. If you want to delete directories that are full, use the deltree command or if you're using Windows 2000 or later use the below example.

rmdir c:\test /s

Windows 2000, Windows XP and later versions of Windows can use this option with a prompt to permanently delete the test directory and all subdirectories and files. Adding the /q switch would suppress the prompt.

COPY: Copy file

copy *.* a:

Copy all files in the current directory to the floppy disk drive.

copy autoexec.bat c:\windows

Copy the autoexec.bat, usually found at root, and copy it into the windows directory; the autoexec.bat can be substituted for any file(s).

copy win.ini c:\windows /y

Copy the win.ini file in the current directory to the windows directory. Because this file already exists in the windows directory it normally would prompt if you wish to overwrite the file. However, with the /y switch you will not receive any prompt.

copy myfile1.txt+myfile2.txt

Copy the contents in myfile2.txt and combines it with the contents in myfile1.txt.

copy con test.txt

Finally, a user can create a file using the copy con command as shown above, which creates the test.txt file. Once the above command has been typed in, a user could type in whatever he or she wishes. When you have completed creating the file, you can save and exit the file by pressing CTRL+Z, which would create ^Z, and then press enter. An easier way to view and edit files in MS-DOS would be to use the edit command.

RENAME: Rename a file

rename c:\chope hope

Rename the directory chope to hope.

rename *.txt *.bak

Rename all text files to files with .bak extension.

rename * 1_*

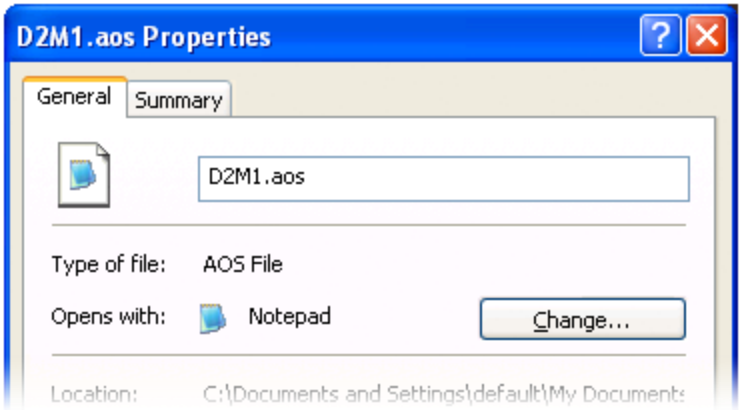
Rename all files to begin with 1_. The asterisk (*) in this example is an example of a wild character; because nothing was placed before or after the first asterisk, this means all files in the current directory will be renamed with a 1_ in front of the file. For example, if there was a file named hope.txt it would be renamed to 1_pe.txt.

Run Script Anywhere

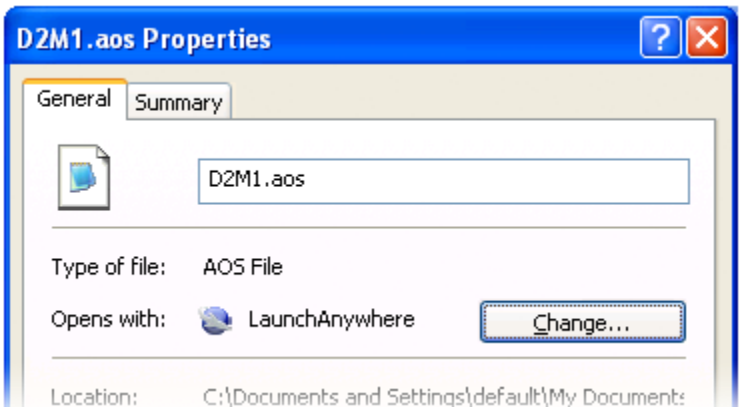
Configuring a Windows Computer

Open your **.aos** file in Notepad. This, by default, will occur when you double-click a script. If it does not, then open Notepad, Press [**Ctrl-O**], and navigate to you .aos script.

Using any .aos file, right-click on the file. In the contextual menu select **Properties**. When the dialog box appears select the [**Change...**] button.



In the **Open With** dialog box the ScriptRunner program may not be listed in the **Other Programs** list so you will have to locate it. Click the [**Browse...**] button and navigate to the location of your file. It can be found in the root directory of the AutoMap folder.

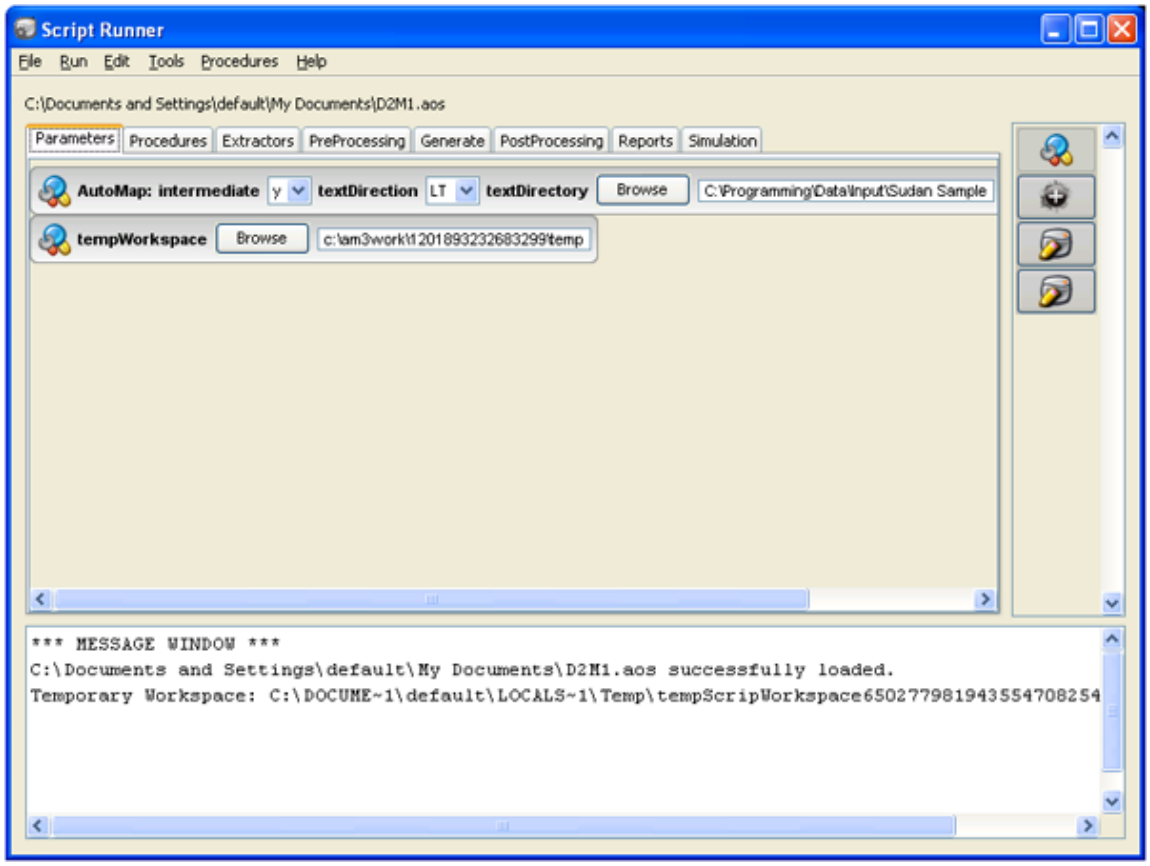


The properties window will now identify **LaunchAnywhere** as the program to run .aos files. Do not worry that it does not say **ScriptRunner**.

The icons used for the .aos files will change to reflect that when you double-click on then a different program will be run. The .aso files should look like the icon to the right. You can now double-click on any of the .aso files which will now launch ScriptRunner and run the script.

The top icon is from Windows XP and the bottom icon is from Windows 7.

You can now use the ScriptRunner to run your .aos scripts. Select you file by using the Run menu.



05 OCT 11

Data-to-Model

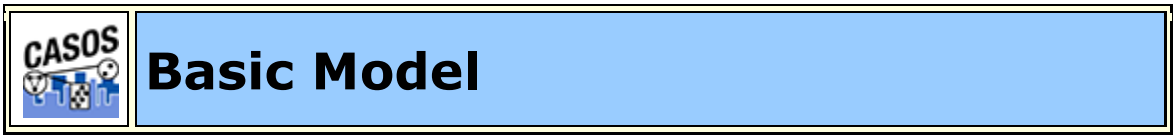
What is Data-to-Model?

Data-to-Model (D2M) is a heuristic procedure for extracting network data from a set of source texts and subsequently analyzing it; source material may include but is not limited to newspapers, magazines, tribune reviews, works of prose, and email. Automap is used to clean and extract the networks from the texts, which can then be analyzed in ORA. (Carley et al., 2010) The analysis made from the report generated by ORA

helps identify the influential people, the intervening agents and implicated locations.

In addition, it helps forecast a situation and identify key actors (Carley et al., 2010). This analytical document is important for policy makers and people interested in the social, political and structural evolution of a situation. Data-to-Model has been used in the case of Sudan Conflict, Singapore and Haiti. There are three degrees of modal that can be obtained: **Basic Model, Refined Model and Advanced Model.**

8 APR 11



Basic Model (AutoMap)

The first step to construct a model is to develop a basic model from texts. This basic model will use the most appropriate routines and techniques and databases requiring limited interaction from the user. The networks in the basic model include **a concept network and a semantic network**.

To reduce the number of concepts in this network, especially multiple concepts that express an identical meaning, a **depluralization thesauri** is constructed focusing on nouns and verbs to take these concepts to their base form such as present tense and singular form.

Established databases are used to identify and process known entities such as the **names of countries and major cities** as well as the names of current and recent world leaders.

Procedures

Step 1 : Create a Project Directory

Prior to uploading your data, you need to create a workspace (folder) where all your input and output files will be stored. This helps in organizing your files and in preventing any loss. You may copy in some standard files such as Generic Delete File,

Standard Thesauri. Information in the Generic Delete List consists of things that have been considered irrelevant in precedent encounters and therefore saved into a Delete List. There may be information that are already pre-existing in our data base that you want to make use of.

Example : CASOS Group has standard thesauri that contains some pre-defined knowledge.

Step 2 : Import your text files into AutoMap

When you click on the **File > Import Text Files** you will be prompted to choose the files you want to upload from your directory. Your files will be uploaded as they are, however you may change the text settings. AutoMap can guess your files encoding but it is not all accurate. It is better to choose your text encoding if you know it before resorting to the automated choice. Other languages settings will require you to change the font to be able to read it. Since your files are from multiple sources, it is certain that your files have different encoding. To facilitate this you can save your files in word as a text file. Due to the huge number of files it takes a lot of time to identify the encoding for each individual text.

Step 3 : Cleaning the Text

There are many concepts and words and structures that are part of your data set but which are not necessary for the purpose of your project. Therefore this need to be deleted from your text. This is selected under **Preprocess > Perform All Cleaning**. This cleaning gets rid of extra whitespace, fixes common typos, coverts British to American spelling, and expands contractions and abbreviations. You can actually perform all this at once but if you do not wish to remove extra space for example you can do the manual cleaning for each step. The individual functions can be found under **Preprocess > Text Cleaning**.

NOTE : This cleaning doesn't affect the meaning of your text.

Step 4 : Generating some thesauri

You are generating these thesauri early because they rely on the **Part of Speech**, thus very important. Before you start manipulating your files it is important to extract the essential

knowledge. Proper nouns and verbs have a tremendous importance in your project. For any generation procedures select them from the **Generate Menu** and scroll down to what you want to generate from the menu list.

a) Suggested Names Thesauri

Generate the **Named Thesauri** from the data. This is automatically executed and it is saved in the project folder where the user can review it when necessary. To generate the Names thesauri, select **Generate > Named Entities**. The Named Entities Thesauri consist of **names of agents, organizations and locations**. It will be saved in the project directory in Standard Format. You may open it in Excel or Word to edit. You may delete some entries that you deem irrelevant and add some from other sources.

NOTE : Gazetteer is a source where you can obtain names of locations to expand your thesauri.

You may generate multiple Name Thesauri and compare them. This thesauri has everything that the part of speech has identified as proper noun. There may be inaccurate facts where some parts of speech are mistakenly identified as nouns but that are not. It is important to know that the system is giving you more information instead of less because it is easy to go through and delete what you don't want than add new things. Factual errors stem also from the structure of the text itself. For instance **Sudan Bishop accuses Oil Companies**, which has been identified by the computer as a name because most of it starts with a capital letter and the computer is not able to differentiate nouns from other parts of speech not because of the way it is presented in the text. The system also gives you a guess of ontological classes (organization, location, agents).

In addition, in AutoMap there is a **Location Distillation** that gives you a thesauri based on the location you specify. If you specify the name of the location the system will suggest all synonyms and spelling variants for that location. You may use those to expand your thesauri. In default everything is classified as agent.

b) Generate a Suggested MetaNetwork Thesauri

From the menu select **Generate > Suggested Metanetwork Thesauri**. This assigns an ontological class for each individual concept. It tells whether this concept is an agent, location, source, or any other category. This automatic categorization is not always right, therefore you may find some obvious proper names classified as locations. This may be due to the structure of the text. The good thing is that you, as a user, can access this thesauri from your project folder and change some classification that you think are not right or just for the purpose of this particular project you may want to classify some obvious names of locations as source or agent.

Example : United States of America is a location but it can also be considered as agent in some cases where the United States Government has taken some actions.

c) Depluralization Thesauri

Depluralization is the elimination of plurals forms which consequently reduces the verbs or nouns to its base form. It uses the part of speech. The Depluralization Thesauri is a list of nouns and verbs in their base forms automatically generated by the Data to Model wizard and saved in the project folder where it can be reviewed anytime. This also includes detensying (reducing verbs to their base forms).

From the main menu select **>Generate > Generalization Thesauri > Context- Stemming Thesaurus**. This procedure applies stemming to nouns and verbs. Proper nouns will remain unchanged. Exception has been drawn on proper nouns because the stemming system doesn't work well with proper nouns.

Example : CASOS becomes CASO which really reduces the meaning or may even refer to something else than what was intended.

We also focus on nouns and verbs because they are the most important part of speech you use in your thesauri. Sometimes due to the text there are some irregularities, irrelevant entries can get involved but you can access it and edit from your directory folder.

NOTE : These are in Master format which refers to the four format thesauri. See **Master Format page** for more information.

After generating the Context-Sensitive Thesaurus, apply it to your text. All nouns and verbs except proper nouns will be reduced to their base forms. It will depluralize and detense most nouns and verbs.

Data Preparation

At this stage you have already extracted the thesauri that rely on the part of speech. You can now manipulate your texts knowing that you have already obtained some essential information.

Step 5 : Pronoun Resolution

The pronoun resolution is done from the Preprocess tab>Text Preparation>pronoun resolution. It consists of replacing all pronouns with their relative nouns.

```
Example: John went to the bakery, he bought some
bread
```

The **he** will be replaced with **John**.

Some pronouns will still remain after this process; all remaining pronouns will be automatically deleted. It also removes prepositions, verbs of noise (verbs of being, verbs of helping), converts all concepts to lower case except proper nouns and names of Organizations and also converts N-grams (two word concepts that appear meaningfully together). Their separation distorts the meaning.

Example : The terms **civil war, white house, United States** have a commonly known meaning being put together. However, each word taken away will have a completely difference meaning. So by converting n-grams, the wizard associates those concepts.

NOTE : It is important that you lower case your text with caution because it may change the ontological classes of the

concepts. Not everything needs to be lower case, especially proper nouns.

Step 6 : Apply the Delete List

To apply Delete List select from the menu **Preprocess > Text Refinement > Apply Delete List**. Applying the Delete List will remove all concepts already in the Delete List and a Filtered List of concepts will be generated. This application should be **Rhetorical** which replaces all deleted concepts with **XXX**. Whereas the **Delete** option will simply apply deletion.

There are cases where you don't want to use the standard delete because some texts are very sensitive. This is not an issue in media files because the idea can still be inferred even after deletion of noise words. However, court documents are very word sensitive, deleting prepositions like **the or a** may drastically change the meaning of that word. A good example is:

```
He shot him with a gun

He shot him with the gun

He shot him with XX gun
```

These sentences have different meaning and may affect the meaning and purposes intended in a court.

Step 7 : Merging

Merge all Depluralization Thesauri, Named Entities Thesauri, and Suggested Thesauri to form a project based thesauri. From the main menu select **Procedures > Master Thesauri procedures > Master Thesauri Merge**. You may also view this list and edit it to fit your project. You are merging them together to have a more expanded knowledge about agents, locations, sources. You will have all this information organized in one file.

Under **Procedures > Thesuari Procedures** you can change your thesauri from Master format to Standard format or visa versa. Under Thesauri Procedures you can also merge thesauri together by specifying the change thesauri and the standard thesauri. You have access to this merged format and may edit it to your liking. Merging the thesauri can also be done manually by copying and pasting them together. At this stage you are the

master of your project therefore you can choose to manually modify your thesauri and tailor it your project. However, you can also execute this automatically under Master Thesauri procedures and in case of conflicts the system will identify the conflict and will prompt you to choose one preference.

Step 8 : Generalization Using Name Thesauri (project thesauri)

This procedure is under the drop list of the Preprocessing tab. Go to Preprocess>Text Refinement>Apply Generalization Thesauri. At the prompt, select the name thesauri and apply it. People, locations and things can have various names. Creating general thesauri will consolidate each of these names into a uniform name (See AutoMap Help). Below is an example: Concept, Key Concept

```
Barack Hussein Obama, Barack_Obama

United States, United_States_of_America

USA, United_States_of_America
```

Step 9 : Generate a Concept List with MetaNetwork Tag

From the main menu select **Generate > Concept List > Concept List with MetaNetwork Tags**. This extracts a list of concept from your data using your Standard Thesauri. From the menu select **Procedures > Thesaurus Procedures > Convert Master Thesauri to MetaNetwork Thesauri**. Use this concept list for any future modifications.

Step 10 : Create an Uncategorized Thesauri

From the main menu select **Generate > Concept List > Concept List (Per Text)**. The Concept List will be extracted from your data. Those concepts have not yet been categorized, in other words there are yet unknown. An ontology will be automatically for each concept found based on part of speech. Merge these concepts to your already existing thesauri. Verbs are classified as tasks and all remaining concepts (except nouns and verbs) as knowledge. You have already classified nouns as agents and location. You can merge this one back to your project thesauri.

Step 11 : Generate DyNetML (Use Metanetwork)

It is now time to generate a DyNetML file. From the main menu select **Generate > Metanetwork > MetaNetwork DyNetML [(Per Text) / (Union Only)]**. The DyNetML is the model you have been aiming for by refining and manipulating your data. You will be prompted to choose a DyNetML from each text or a Union DyNetML which will create one file using concepts from all files. Choose a **window size** based on your average sentence length in order to have an adequate view of your DyNetML. Windows size 8 is often used.

Step 12 : Start ORA

Load these files already saved in your project folder to ORA. Generate key entity report using union. Upon generation of the key entity report, you may review the report. If your report appears to be lacking for the analysis of your project, you may always go back to the thesauri and tailor it to your project purposes. Depending on your satisfaction of the results generated by ORA you can always go back to refine your Project Thesauri and generate a new DyNetML and then a new ORA report. It is the refinement process.

2 JUN 11

Refined Model

Refined model

The refined model allows the user to evaluate the automated choice selections. For instance, in the depluralization thesauri concepts are taken to their base form. This technique uses part of speech analysis to find only nouns and verbs, specifically excluding proper nouns. However, an occasional proper noun may be identified as a common noun especially in the case of incorrect grammar usage in texts.

The names thesauri use proper names to identify instances of agents. While it is common for the proper names found in a text corpus to refer to an agent, a proper name could refer to an

organization. The names thesauri would be reviewed to change the categorization to organization where appropriate. To review the names thesauri you can access it from your project folder. The category agent can be substituted with organization or location. As a user, if you feel that the entry does not correspond to agent, organization, or location, one of the other categories can be used such as event, resource, knowledge, or task. If an entry does not fit any of those categories the entry can be deleted from the thesauri.

7 APR 11

Advanced Model

Advanced Model

In the advanced Model the user is well acquainted with the data and with the procedures. Therefore, you may use more expertise to execute procedures without the wizard because you now understand the purpose and the under-belly of the data to Model wizard.

2 JUN 11

Analysis

Analysis

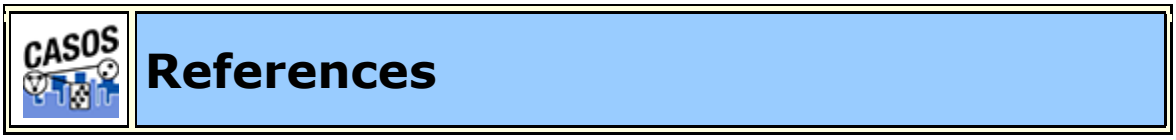
This steps calls for your knowledge of the subject you are dealing with and also knowledge about the actor's level measures and the network level measures. This includes but is not limited to degree centrality, hub centrality, click counts, authority centrality etc (ORA Glossary, 2010). Prior to the analysis, you have already obtained your Model which is the DyNetML. This is only the analytical part, no more automated procedure is involved, and this should be done after all satisfactory refinements.

References

Carley, K.M., Reminga J., Storrick J., and Columbus D., 2010, "ORA User's Guide 2010,"Carnegie Mellon University, School of Computer Science, Institute for Software Research, Technical Report, CMU-ISR-10-120.

Carley, K.M., Columbus D., Bigrigg M. and Kunkel F., 2010 "AutoMap User's Guide 2010,"Carnegie Mellon University, School of Computer Science, Institute for Software Research, Technical Report, CMU-ISR-10-121.

Carley, K.M; Tambayong. L (2010). Political Networks of Sudan: A two-Mode Dynamic Network Text Analysis. Carnegie Mellon University of Pittsburgh, CASOS group.



References

Borgatti, S. P., M. G. Everett, and L. C. Freeman. (2002). UCINET for Windows, Software for Social Network Analysis: Analytic Technologies, Incorporated.

Burkart, Margaret. (1997). Thesaurus. In Marianne Buder, Werner Rehfeld, Thomas Seeger & Dietmar Strauch (Eds.), *Grundlagen der praktischen Information und Dokumentation: Ein Handbuch zur Einführung in die fachliche Informationsarbeit* (4th ed., pp. 160 - 179). München: Saur.

Carley, Kathleen M. (1993). Coding Choices for Textual Analysis: A Comparison of Content Analysis and Map Analysis. *Sociological Methodology, 23*, 75-126.

Carley, Kathleen M. (1993). Content Analysis. In R.E. Asher & J.M.Y. Simpson (Eds.), *The Encyclopedia of Language and Linguistics* (Vol. 2, pp. 725-730). Edinburgh, UK: Pergamon Press.

Carley, Kathleen M. (1994). Extracting Culture through Textual Analysis. *Poetics, 22*, 291-312.

Carley, Kathleen M. (1997). Extracting Team Mental Models Through Textual Analysis. *Journal of Organizational Behavior, 18*, 533-538.

Carley, Kathleen M. (1997). Network Text Analysis: The Network Position of Concepts. In Carl W. Roberts (Ed.), *Text Analysis for the Social Sciences: Methods for Drawing Statistical Inferences from Texts and Transcripts* (pp. 79-100). Hillsdale, NJ: Lawrence Erlbaum Associates.

Carley, Kathleen M. (2002). Smart Agents and Organizations of the Future. In Leah Lievrouw, and Sonia Livingstone (Ed.), *The Handbook of New Media* (pp. 206-220). Thousand Oaks, CA: Sage.

Carley, Kathleen M. (2003). Dynamic Network Analysis. In Ronald Breiger, Kathleen Carley & Philippa Pattison (Eds.), *Dynamic Social Network Modeling and Analysis: Workshop Summary and Papers, Committee on Human Factors* (pp. 133-145). Washington, DC: National Research Council.

Carley, Kathleen M., Diesner, Jana, Reminga, Jeffrey, & Tsvetovat, Maksim. (2007). Toward an Interoperable Dynamic Network Analysis Toolkit. *Decision Support Systems: Special Issue Cyberinfrastructure for Homeland Security, 43*(4), 1324-1347.

Carley, Kathleen M., and David Kaufer. (1993). Semantic Connectivity: An Approach for Analyzing Semantic Networks. *Communication Theory, 3*(3), 183-213.

Carley, Kathleen M., and Michael Palmquist. (1992). Extracting, Representing and Analyzing Mental Models. *Social Forces, 70*(3), 601-636.

Carley, Kathleen M., & Reminga, Jeffrey. (2004). ORA: Organizational Risk Analyzer. Pittsburgh, PA: Carnegie Mellon University, School of Computer Science, Institute for Software Research.

Diesner, Jana, & Carley, Kathleen M. (2004). AutoMap 1.2 : Extract, Analyze, Represent, and Compare Mental Models from Texts. Pittsburgh, PA: Carnegie Mellon University,

School of Computer Science, Institute for Software Research.

Diesner, Jana, & Carley, Kathleen M. (2005, April 21-23). *Exploration of Communication Networks from the Enron Email Corpus*. Paper presented at the SIAM International Conference on Data Mining: Workshop on Link Analysis, Counterterrorism and Security, Newport Beach, CA.

Diesner, Jana, & Carley, Kathleen M. (2005). Revealing Social Structure from Texts: Meta-Matrix Text Analysis as a novel method for Network Text Analysis. In V.K. Narayanan & D.J. Armstrong (Eds.), *Causal Mapping for Information Systems and Technology Research: Approaches, Advances, and Illustrations* (pp. 81-108). Harrisburg, PA: Idea Group Publishing.

Diesner, Jana, Carley, Kathleen M., & Katzmair, Harald. (2007, May 1-6). *The morphology of a breakdown. How the semantics and mechanics of communication networks from an organization in crises relate*. Paper presented at the XXVII Sunbelt Social Network Conference, Corfu, Greece.

Diesner, Jana, Kumaraguru, Ponnurangam, & Carley, Kathleen M. (2005). *Mental Models of Data Privacy and Security Extracted from Interviews with Indians.* Paper presented at the 55th Annual Conference of the International Communication Association (ICA), New York, NY.

Diesner, Jana, & Stuetzer, Cathleen. (2008, July 24). *Relationen finden/Finding Relations.* Paper presented at the Kunstsammlungen Chemnitz, Chemnitz Art Collections.

Jurafsky, Daniel, & Marton, James H. (2000). *Speech and Language Processing.* Upper Saddle River, New Jersey: Prentice Hall.

Kaufer, David, and Kathleen M. Carley. (1993). Condensation Symbols: Their Variety and Rhetorical Function in Political Discourse. *Philosophy and Rhetoric, 26*(3), 201-226.

Klein, Harald. (1996). Classification of Text Analysis Software. In Rudiger Klar & Otto Opitz (Eds.), *20th Annual Conference of the Gesellschaft fÃ¼r Klassifikation e.V.* (pp. 255-261).

University of Freiburg: Springer. Krovetz, Robert. *Word Sense Disambiguation for Large Text Databases.* Unpublished PhD Theis, University of Massachusetts, 1995.

Magnini, Bernardo, Negri, Matteo, Prevete, Roberto, & Tanev, Hristo. (2002). A Wordnet-based Approach to Named-Entites Recognition *SemaNet'02: Building and Using Semantic Networks* (pp. 38-44). Taipei, Taiwan.

Mrvar, Andrej. (2004). Measures of Centrality and Prestige, from http://mrvar.fdv.uni-lj.si/sola/info4/uvod/part4.pdf

Palmquist, Michael, Kathleen M. Carley, and Thomas Dale. (1997). Two applications of automated text analysis: Analyzing literary and non-literary texts. In C. Roberts (Ed.), *Text Analysis for the Social Sciences: Methods for Drawing Statistical Inferences from Texts and Transcripts* (pp. 171-189). Hillsdale, NJ: Lawrence Erlbaum Associates.

Popping, R., & Roberts, C.W. (1997). Network Approaches in Text Analysis. In R. Klar & O. Opitz (Eds.), *20th Annual Conference of the Gesellschaft fÃ¼r Klassifikation e.V.* (pp. 381-898). University of Freiburg: Springer.

Porter, M.F. (1980). An algorithm for suffix stripping. *Program, 14*(3), 130-137.

Shannon, Claude E., & Weaver, Warren. (1949). *The Mathematical Theory of Communication.* Urbana, IL: University of Illinois Press.

Tsvetovat, Maksim, Reminga, Jeffrey, & Carley, Kathleen M. (2004). DyNetML: Interchange Format for Rich Social Network Data. Pittsburgh, PA: Carnegie Mellon University, School of Computer Science, Institute for Software Research. From http://reports-archive.adm.cs.cmu.edu/anon/isri2004/abstracts/04-105.html

Wasserman, Stanley, & Faust, Katherine. (1994). *Social Network Analysis: Methods and Applications.* Cambridge: University of Cambridge Press.

Zuell, Cornelia, & Alexa, Melina. (2001). Automatisches Codieren von Textdaten. Ein Ueberblick ueber neue Entwicklungen. In Werner Wirth & Edmund Lauf (Eds.), *Inhaltsanalyse - Perspektiven, Probleme, Potenziale* (pp. 303-317). Koeln: Herbert von Halem.